

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

OPTICKÉ POLE A SYNTÉZA HOLOGRAMŮ

BAKALÁŘSKÁ PRÁCE

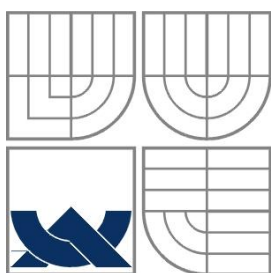
BACHELOR'S THESIS

AUTOR PRÁCE

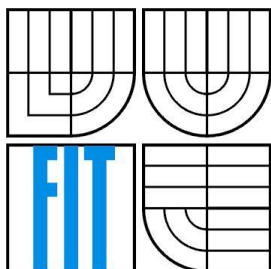
AUTHOR

MARIÁN ŠULGAN

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

OPTICKÉ POLE A SYNTÉZA HOLOGRAMŮ

OPTICAL FIELD AND HOLOGRAM SYNTHESIS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

MARIÁN ŠULGAN

VEDOUCÍ PRÁCE
SUPERVISOR

prof. Dr. Ing. PAVEL ZEMČÍK

BRNO 2014

Abstrakt

Cílem této práce je implementovat syntézu optického pole vybranou metodou v různých podobách, porovnat je a výsledek potvrdit rekonstrukcí hologramu vyrobeného na základě vypočteného optického pole. Z důvodu náročnosti je výpočet urychlený pomocí později popsané optimalizace algoritmu, pomocí paralelizace a využitím technologie C++ AMP, díky které je možné zužít výpočetní sílu GPU. Práce na začátku vysvětluje známé fyzikální principy holografie, následuje návrh řešení stanovených problémů. Závěrečná část popisuje konkrétní řešení v podobě konsolové aplikace naprogramované v C++, způsob testování, získané výsledky a jejich zhodnocení. Konec práce tvoří finální shrnutí a uvažování nad možným pokračováním a zlepšením do budoucna.

Abstract

The main focus of this work is to implement appropriate chosen method of optical field synthesis in several forms, compare their results and confirm its validity by reconstruction of hologram generated from calculated optical field. For its complexity and demanding calculations, various accelerations are employed - linear approximation, parallelization, using technology C++ AMP, which facilitates opportunity to use GPU for massive parallelism. The work begins with explanation of basic holography principles followed by design of application offering solutions to set problems. The final part explains the implementation of application programmed in C++, testing process, gained results and their evaluation. The text ends with final conclusions regarding completion of set tasks and possible future expanding or improving.

Klíčová slova

Holografie, optické pole, hologram, wireframe, point cloud, OpenMP, C++ AMP, GPU

Keywords

Holography, optical field, hologram, wireframe, point cloud, OpenMP, C++ AMP, GPU

Citace

Šulgan Marián: Optické pole a syntéza hologramů, bakalářská práce, Brno, FIT VUT v Brně, 2014

Optické pole a syntéza hologramů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením prof. Dr. Ing. Pavla Zemčíka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Marián Šulgan
20. května 2014

Poděkování

Na tomto místě bych se chtěl poděkovat vedoucímu práce prof. Dr. Ing. Pavlu Zemčíkovi za jeho odborné vedení a ochotu vytrvale pomáhat s tvorbou této práce. Též bych rád poděkoval Ing. Petru Lobazovi ze Západočeské univerzity v Plzni za trpělivé zodpovídání mnoha dotazů o holografii a poskytnutí kvalitních studijních materiálů, a Ing. Michalu Seemanovi, Ph. D., za úspěšnou rekonstrukci hologramu.

© Marián Šulgan, 2014

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod.....	2
2	Holografia	3
2.1	Základné princípy holografie.....	3
2.2	Optická holografia	8
2.3	Digitálna holografia	15
3	Techniky a aplikácie	19
3.1	SDK Holography Toolkit	19
3.2	OpenMP.....	19
3.3	C++ AMP	21
3.4	„Wireframe“ a „point cloud“	22
4	Zhodnotenie stavu a plán práce.....	23
4.1	Zhodnotenie súčasného stavu	23
4.2	Návrh riešenia aplikácie	24
5	Implementácia.....	27
5.1	Celková koncepcia aplikácie	27
5.2	Významné časti riešenia	28
5.3	Testovanie aplikácie a výsledky	33
6	Záver	39

1 Úvod

Zrak a s ním spojené vizuálne vnímanie je nepochybne naším najsilnejším zmyslom. Ľudia od nepamäti využívajú obrazovú tvorbu na predávanie informácií či vyjadrenie svojich pocitov. Postupom času sme od jaskynných malieb cez rôzne maliarske techniky dospeli až do bodu, kedy sme boli schopní obraz nielen vytvoriť, ale aj zaznamenať. V roku 1920 tak vzniká fotografia. Fotografia má svoje nespochybniteľné čaro aj miesto v spoločnosti, no nedokáže zaznamenať priestor, ktorý fotografujeme v jeho hĺbke. Pracuje len v dvoch rozmeroch, čo na jednej strane prináša krásu práce s ňou, napríklad zaostrenie na určitý objekt, ale na druhej strane vytvára určitý priestor a smer, ktorým sa ďalej uberať. Ako čitateľ môže tušiť, skutočne sa vývoj týmto smerom aj uberal. V roku 1948 je maďarským fyzikom Denisom Gaborom pri výskume zdokonalenia elektrónového mikroskopu vynájdená holografia. Išlo však len o teoretický základ, praktické využitie, experimentovanie a ďalší vývoj stagnovali až do vynájdenia vhodného zdroja svetla – laseru.

Samotné slovo holografia pochádza z gréckeho *holos*, čo znamená celý, a *graphie*, čo značí zápis. Dá sa o nej v určitom zmysle hovoriť ako o vyspelej forme fotografie, hoci všeobecne sa zaužívali rôzne mýty. Umelecký žáner science-fiction prispel k tejto mystifikácii, keď vo viacerých filmoch a knihách prezentuje hologramy – výsledky holografie – ako objekty v priestore bez nosiča. Toto v skutočnosti nie je pravda, keďže hologram, ako bolo spomenuté, je skôr bližší fotografii, čiže je zaznamenaný na určitom nosiči. Ak sa pýtate, čo to teda vlastne je a aký je jej prínos, pokúsím sa odpovedať. Podstata spočíva v tom, že holografia je schopná zaznamenať celú hĺbku priestoru, nenastáva efekt rozostrenia. Je skutočným záznamom priestoru. Teoreticky, pri dokonalom zázname, nie sme schopní rozoznať skutočný objekt od pohľadu na hologram. Z toho plynie aj fakt, že pri pohľade na hologram z rôznych uhlov sa zaznamenaný objekt javí otočený, tak isto, ako keby skutočne existoval v priestore.

Pri narastajúcich nárokoch spotrebiteľov je zároveň potenciálne vhodnou náhradou súčasných 3D technológií, ktoré pracujú s nedokonalosťou ľudských očí. Je nutné mať nasadené špeciálne okuliare, čo je výrazne obmedzujúce a nepohodlné. Niektorí ľudia si na to zvykajú ťažko, vyhýbajú sa tejto technológii. Holografia tieto problémy nemá, nevyžaduje špeciálne zariadenie na strane užívateľa.

Na druhej strane má tiež viacero nevýhod, ktoré v súčasnosti bránia väčšej expanzii. Pri digitálnej holografii sú výpočty veľmi náročné a pri holograme, ktorý by bol dosť veľký a dosť kvalitný na plnohodnotný zážitok pre človeka, je nevyhnutné sa zaoberať výraznou optimalizáciou, pretože výkon bežných dnešných počítačov nepostačuje. Ak by sa jednalo o holografické video, nároky ešte stúpnu – hoci, vedcom z MIT sa pomocou niekoľkých trikov podarilo aj toto.

Holografia má nesporne obrovský potenciál. Primárny prínos je pri zobrazovaní informácií v trojrozmernom priestore bez skreslenia, uplatnenie by našla v mnohých oblastiach života. Fascinovaný touto oblasťou som sa rozhodol venovať túto prácu práve téme holografia.

Prvá kapitola je venovaná teoretickému prehľadu holografie, jej základných princípov a rozdeleniu na digitálnu a optickú. V druhej kapitole je popísaných niekoľko možných nástrojov a pojmov, ktoré sa vzťahujú práve k digitálnej holografii a sú použiteľné v kontexte tohto diela. Nasleduje kapitola, ktorá vo svojej prvej časti zhrňa súčasný stav tejto oblasti a snaží sa vytýčiť možné problémy a v druhej navrhuje možné riešenia. Predposledná kapitola sa venuje konkrétnemu implementovanému riešeniu, jeho častiam a spôsobu testovania, overenia výsledkov a diskutuje ich. Záver práce tvorí konečné zhrnutie získaných poznatkov a možnosti ďalšieho pokračovania v budúcnosti.

2 Holografia

Holografia stojí na pomerne komplexných optických zákonoch. Priblíženiu si fyzikálnej podstaty tejto technológie sa venuje kapitola Základné princípy holografie. Z týchto základov vychádza holografia ako taká, či už digitálna alebo optická, a preto budú v spomenutej kapitole uvedené. Stručne vysvetlené budú javy ako vlnový model svetla, interferencia, difrakcia či koherencia. Ďalšie dve kapitoly sa venujú rozdeleniu holografie na optickú a digitálnu s priblížením každej z nich.

Vhodné je ešte podotknúť, že táto práca v žiadnom prípade nie je encyklopedickým prehľadom o holografii. Obsahuje len informácie relevantné pre túto prácu - ide skôr o všeobecný popis niektorých javov v optike pre čitateľa, ktorý nie je odborník v danej oblasti. Obsiahlejší popis by bol samoučelný.

Holografia je vedecká oblasť staršia ako by sa nezaujatému čitateľovi mohlo zdať. Už v roku 1948 jej teoretické základy položil britský fyzik maďarského pôvodu Dennis Gabor (Dénes Gábor), za čo bol v roku 1971 ocenený Nobelovou cenou. Objav bol neočakávaným výsledkom pri pokuse o vylepšenie elektrónového mikroskopu. Prvý zostrojený hologram bol typu „in-line“ a poskytoval iba nekvalitné výsledky. Vývoj na niekoľko rokov stagnoval, až do vynájdenia zdroja dostatočne koherentného svetla – laseru, v roku 1960. Prvý záznam trojrozmerného objektu, významný krok vo vývoji, sa udial v roku 1962 a na svedomí ho mali vedci Emmett Leith a Juris Upatnieks v USA a Yuri Denisjuk v Sovietskom zväze. Išlo o tzv. „off-axis“ hologram. Obom typom, vrátane charakteristiky niekoľkých iných, sa bližšie venuje kapitola Optická holografia.

Myšlienky tvorby hologramov na počítačoch tiež nie sú dielom posledných rokov. Prvé pokusy boli už v roku 1967. Vzhľadom na výkon vtedajších počítačov sa ale značne zabrzdil a pokračoval až v deväťdesiatych rokoch, kedy začala byť výpočtová sila počítačov dostačujúca na potrebné simulácie a numerické výpočty. Tieto myšlienky, spolu s podrobnejším popisom a metódami syntézy hologramov s niektorými ich optimalizáciami, sú uvedené v kapitole Digitálna holografia.

Holografia ako taká je vyspelá forma záznamu obrazu. Ten sa nazýva hologram. Svetlo, rozptýlené objektom, sa zaznamená. Po následnej rekonštrukcii sa obraz nachádza na rovnakom mieste voči záznamovému nosiču, t. j. obraz sa mení na základe pozície pozorovateľa voči hologramu. Fakticky ide o záznam trojrozmerného objektu na dvojrozmerné obrazové médium. Vo svojom princípe sa ponáša na fotografiu. Rozdiel je v zaznamenávaných veličinách, fotografia zaznamenáva len intenzitu svetla, pričom na holograme je zaznamenaná okrem intenzity aj fáza svetla odrazeného od predmetu. Práve informácia o fáze je kľúčom k záznamu v troch rozmeroch.

Pre úplnosť je potrebné dodať, že holografia je v skutočnosti rozsiahla vedecká oblasť. Podľa jej aplikácie ju môžeme rozdeliť do viacerých kategórií, napríklad oblasť optických počítačov či nedeštruktívneho testovania (NDT). V tejto práci sa však pracuje s pojmom holografia v zmysle trojrozmerného zobrazovania, ako bolo spomenuté vyššie.

2.1 Základné princípy holografie

Táto kapitola sa bude venovať popisu fyzikálnych javov a s nimi spojených zákonov, ktoré sa uplatňujú v optike a tvoria základ pre pochopenie fungovania holografie. Patria sem najmä javy ako interferencia a difrakcia, vlnenie, či koherentnosť zdroja svetla. Každý bude podrobnejšie popísaný slovné aj vhodnými rovnicami a vzorcami.

Vlnový model svetla

Svetlo, jeho vlastnosti a spôsob šírenia môže byť popísaný pomocou vln, lúčov alebo kvánt. Kvantová optika sa zaoberá dejmi, pri ktorých sa prejavuje kvantový charakter svetla, t. j. na úrovni jednotlivých častíc – fotónov. Na druhej strane, pomocou lúčov sa popisuje chovanie svetla na úrovni ľudského vnímania, napríklad javy ako odraz a lom. Obidva tieto modely zlyhávajú pri vysvetlení interakcie svetla s objektmi veľkosti porovnateľnej s vlnovou dĺžkou svetla, kvantová optika je príliš „zblízka“, vlnová príliš „zďaleka“. Keďže holografia sa zaoberá práve interakciou na úrovni mikroskopických objektov, využíva práve vlnovú optiku [6].

Vlna je vo fyzike definovaná ako proces šírenia elektromagnetickej poruchy prostredím. Pozostáva z časovo premenného magnetického a elektrického poľa. Popisujú ju Maxwellove rovnice a pri splnení určitých predpokladov o vlastnostiach prostredia je popísaná rovnicou skalárnej vlny:

$$\nabla^2 u(p, t) - \frac{n^2}{c^2} \frac{\partial^2 u(p, t)}{\partial t^2} = 0, \quad (2.1)$$

kde $u(p, t)$ je porucha na pozícii p v čase t v prostredí, ktoré má index lomu n . Porucha sa šíri prostredím rýchlosťou c/n , pričom c je rýchlosť svetla. Ako je vyššie spomenuté, táto porucha sa nazýva vlna. V ďalšom texte bude používané toto pomenovanie [7].

Vlastnosťou vlny je jej vlnová dĺžka, ktorá je definovaná ako $\lambda = v/f$, kde v je rýchlosť šírenia vlny v [m/s] a f je frekvencia vlnenia v [Hz]. Vyjadruje vzdialenosť medzi opakujúcimi sa periódami vlnenia. Pre holografiiu je najzaujímavejšie jednofarebné, tzv. monochromatické svetlo, ktoré obsahuje iba vlny rovnakej vlnovej dĺžky (viď. kapitola Koherencia). Rovnica monochromatickej vlny je:

$$u(p, t) = A(p) \cos[2\pi vt - \phi(p)], \quad (2.2)$$

kde $A(p)$ vyjadruje amplitúdu vlny a $\phi(p)$ fázu vlny na pozícii p v čase t . Symbolom v sa označuje optická frekvencia. Táto rovnica sa pre jednoduchšiu prácu zvykne vyjadrovať v exponenciálnom tvare:

$$\begin{aligned} u(p, t) &= \Re\{\tilde{u}(p) \exp(-j2\pi vt)\} \\ &= \Re\{\tilde{u}(p) \exp(-j\omega t)\}, \end{aligned} \quad (2.3)$$

kde zápis $\Re\{\dots\}$ vyjadruje reálnu zložku komplexného čísla a ω uhlovú rýchlosť. Funkcia $u(p, t)$ sa nazýva vlnová funkcia. Člen $\tilde{u}(p)$ je definovaný ako:

$$\tilde{u}(p) = A(p) \exp[j\phi(p)] \quad (2.4)$$

a nazýva sa komplexná amplitúda. Ide o komplexné číslo obsahujúce informáciu o amplitúde aj fáze svetla [2].

Ďalšou dôležitou vlastnosťou je intenzita svetla I . Táto veličina popisuje energiu prenášanú vlnou. Intenzita vlny $\tilde{u}(p)$ sa rovná $|\tilde{u}|^2$, t.j. je vynásobením dvoch navzájom komplexne združených čísel [2]:

$$I = \tilde{u}(p)\tilde{u}^*(p) = |\tilde{u}(p)|^2 \quad (2.5)$$

Typy vln

V prípadoch, ktorá sú pre túto prácu zaujímavé, sa vyskytujú dva elementárne typy vln. Rozdelené sú podľa svojho tvaru, a to na rovinnú a sférickú vlnu.

Vlnoplocha rovinatej vlny je nekonečná rovina. Pre jej komplexnú amplitúdu platí:

$$\tilde{u}(\mathbf{r}) = \tilde{a} \exp(j\mathbf{k} \cdot \mathbf{r}), \quad (2.6)$$

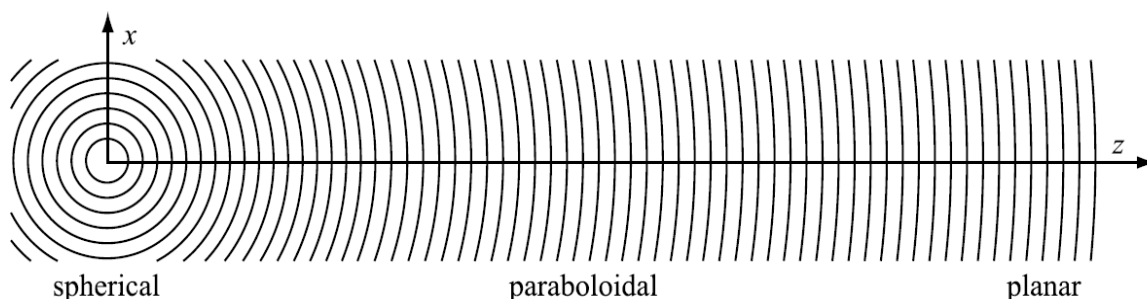
kde $\mathbf{r} = (x, y, z)$ značí polohový vektor, \tilde{a} je komplexná amplitúda popisujúca amplitúdu a fázu v mieste vzniku vlny a symbol \mathbf{k} reprezentuje vlnový vektor, $\mathbf{k} = (k_x, k_y, k_z)$. Jeho dĺžka, $|\mathbf{k}|$, označuje vlnové číslo [7].

Sférická vlna vychádza z bodového zdroja svetla. Jej vlnoplochy sú koncentrické guľové plochy so stredom v mieste tohto bodového zdroja. Komplexná amplitúda sférickkej vlny je:

$$\tilde{u}(r) = \frac{\tilde{a}}{r} \exp(jkr), \quad (2.7)$$

kde $r = |\mathbf{r}|$, t.j. vzdialenosť od zdroja. Na základe členu \tilde{a}/r z uvedeného vzťahu vyplýva, že s rastúcou vzdialenosťou r sa kvadraticky znižuje intenzita vlny, keďže pre intenzitu platí $I = |\tilde{a}|^2$ [7].

Rovinná vlna veľmi úzko súvisí so sférickou. Vo veľmi veľkej vzdialenosti od bodového zdroja sa zo sférickkej vlnoplochy stane plochá, čiže rovinná vlnoplocha. Hovorí sa o nej ako o rovinatej vlně, pretože polomer zakrivenia sa blíži k nekonečnu a úspešne môžeme použiť aproximáciu rovinnou vlnou. Tento postup je značným zjednodušením pri výpočtoch a analýzach [6]. Pre lepšiu ilustráciu vzťahu rovinatej a sférickkej vlny je uvedený obrázok 2.1.



Obrázok 2.1: Prechod sférickkej vlny do rovinatej.¹

Interferencia

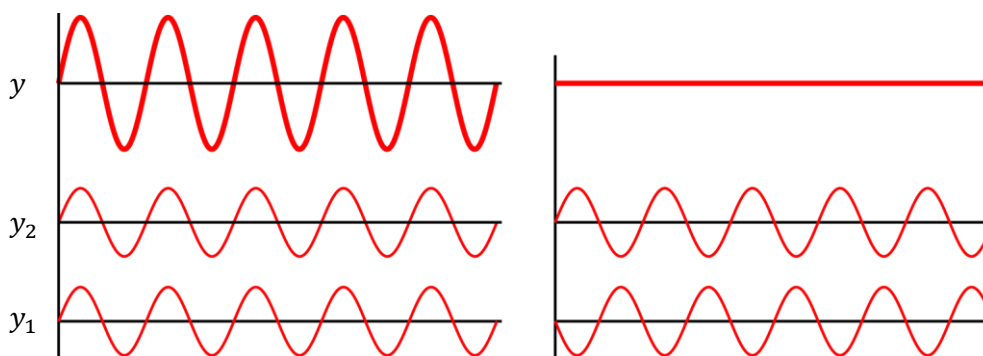
Interferencia je najvýraznejším prejavom vlnovej povahy svetla. Spočíva v tom, že dve alebo viaceré vlny z rôznych zdrojov sa v určitom bode skladajú. V holografii sa využíva pri zázname hologramu.

Je možné hovoriť o dvoch typoch interferencie v závislosti od fázy. Prvým je konštruktívna interferencia. Tá nastáva vtedy, keď fázy v danom bode sú rovnaké alebo je medzi nimi minimálny rozdiel. Potom v tomto bode dôjde k nárastu intenzity. Druhým a opačným prípadom je deštruktívna

¹ Obrázok prevzatý z [6].

² Obrázok upravený, originál prevzatý z: http://en.wikipedia.org/wiki/File:Interference_of_two_waves.svg.

interferencia – ak sú fázy opačné alebo približne opačné, intenzita v danom bode klesne [7]. Oba krajné scenáre interferencie dvoch vln sú znázornené na obrázku 2.2.



Obrázok 2.2: Interferencia (y) dvoch vln (y_1 a y_2). Vľavo úplná konštruktívna, vpravo úplná deštruktívna interferencia.²

Matematický popis je nasledovný. Predpokladajme, že existujú dve vlny popísané ich komplexnou amplitúdou, \tilde{u}_1 a \tilde{u}_2 a obe sú koherentné. Potom výsledok ich interferencie bude vlna $\tilde{u} = \tilde{u}_1 + \tilde{u}_2$. Všeobecne je možné tento vzťah pri rovnakých predpokladoch zjednodušene zapísať ako:

$$\tilde{u} = \tilde{u}_1 + \tilde{u}_2 + \dots + \tilde{u}_n \quad (2.8)$$

Pre výslednú intenzitu I dvoch interferujúcich vln potom platí:

$$\begin{aligned} I &= |\tilde{u}_1 + \tilde{u}_2|^2 \\ &= |\tilde{u}_1 + \tilde{u}_2| |\tilde{u}_1 + \tilde{u}_2|^* \\ &= |\tilde{u}_1|^2 + |\tilde{u}_2|^2 + \tilde{u}_1 \tilde{u}_2^* + \tilde{u}_1^* \tilde{u}_2 \\ &= I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\phi_1 - \phi_2), \end{aligned} \quad (2.9)$$

kde ϕ_1 predstavuje fázu vlny \tilde{u}_1 a ϕ_2 fázu vlny \tilde{u}_2 . Intenzita vypočítaná podľa rovnice 2.9 potom formuje štruktúru nazývanú interferenčný vzor, inými slovami hologram [6].

Koherencia

Ďalším fyzikálnym javom úzko súvisiacim s interferenciou a dôležitým pre záznam hologramu je koherencia. Závisí od nej, či budú jednotlivé interferenčné vzory dostatočne viditeľné, teda aj detekovateľné a zaznamenané na príslušný nosič. Pri tvorbe optického hologramu sa na záznam využíva fotocitlivý materiál. Interferenčný vzor, ktorý sa zaznamenáva, musí byť

y_1 stabilný, t. j. nemeniť sa počas expozície, inak by ho nebolo možné zaznamenať.

Jednoducho keby svetlo nebolo koherentné, vzor by nebol viditeľný [6].

Vlny, ktorých fázový rozdiel v mieste ich skladania sa nemení v čase, nazývame koherentné. V mieste skladania majú rovnaké vlastnosti – vlnovú dĺžku aj počiatočné fázy. V reálnom svete ale

² Obrázok upravený, originál prevzatý z: http://en.wikipedia.org/wiki/File:Interference_of_two_waves.svg.

neexistuje zdroj dokonalého monochromatického svetla, preto sa používa laser, ktorý má k tomu najbližšie – rozsah frekvencií laserového svetla je dostatočne malý.

Matematicky sa viditeľnosť \mathcal{V} takéhoto vzoru dá vyjadriť nasledovne:

$$\mathcal{V} = \frac{2(I_1 I_2)^{1/2}}{I_1 + I_2} \cos(\psi), \quad (2.10)$$

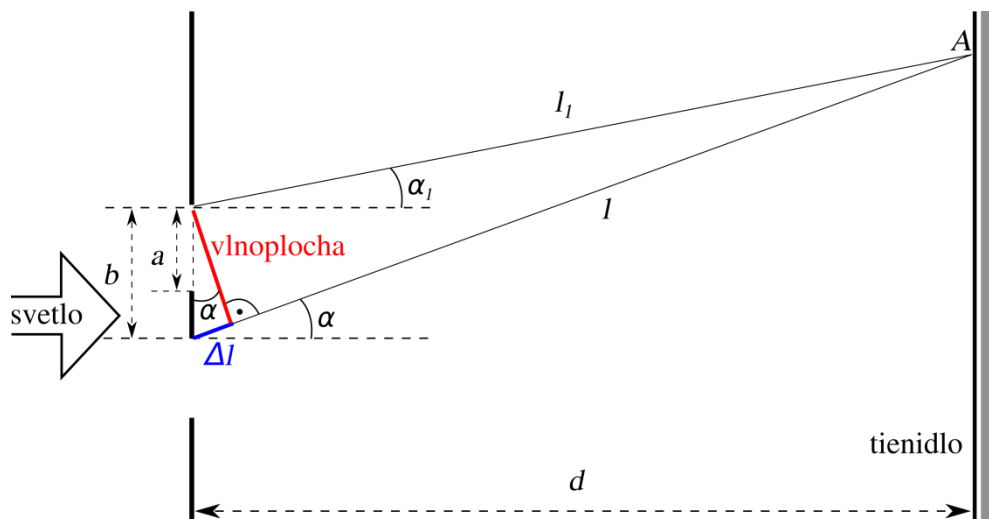
kde I_1 a I_2 sú intenzity dvoch ideálnych koherentných zdrojov svetla tvoriacich interferenčný vzor o intenzite I . Zároveň ψ vyjadruje uhol medzi elektrickými vektormi týchto vln, čím reprezentuje polarizáciu. Za povšimnutie stojí, že viditeľnosť \mathcal{V} bude nulová, ak bude uhol medzi vektormi násobkom $\pi/2$, čiže budú navzájom kolmé [2].

O koherencii svetelného zdroja sa dá hovoriť v dvoch rozmeroch. Prvým je priestorová koherencia, o ktorej hovoríme v prípade vlnenia z dvoch rôznych svetelných zdrojov. Prakticky vyjadruje schopnosť detailného zaostrenia laseru. Druhým rozmerom je časová koherencia – v prípade, že vlnenia vychádzajú z rovnakého miesta, ale v rozdielnom čase a vyjadruje mieru monochromatickosti svetla [1].

Dôležitým faktom je, že ak je svetlo koherentné aj priestorovo aj časovo, je možné v rovnici 2.3 zanedbať časovú zložku, t. j. člen $-j\omega t$. Vďaka tomu môže byť monochromatické svetlo plne charakterizované komplexnou amplitúdou [7].

Difrakcia

Difrakcia je ďalší z dôsledkov interferencie svetla. Taktiež sa mu vraví ohyb svetla. Jednoducho by sa dalo povedať, že difrakcia je odchýlka od priamočiareho šírenia svetla, ale na rozdiel od odrazu alebo lomu je špecifická tým, že vlnenie sa dostáva do oblasti geometrického tieňa. Nastáva na akejkoľvek prekážke, ktorej rozmery sú porovnateľné s vlnovou dĺžkou dopadajúceho svetla. Najzaujímavejším typom prekážky pre holografiiu je tzv. štrbina, čiže otvor so šírkou porovnateľnou s vlnovou dĺžkou svetla [8].



Obrázok 2.3: Difrakcia na dvojici štrbín.³

³ Obrázok upravený, originál prevzatý z [8].

Nech existuje koherentná rovinná vlnu, ktorá dopadá kolmo na rovinu štrbiny. Potom z jej vlnoplochy môže cez štrbinu prejsť len časť. Ohyb je spôsobený Huygensovým princípom. Je to jav, kedy každý bod vlnoplochy možno považovať za zdroj sekundárnej svetelnej vlny. Z toho plynie, že okraj štrbiny sa stane tiež bodovým zdrojom a do oblasti geometrického tieňa sa dostane svetlo [1].

Doteraz bol popisovaný prípad jednej štrbiny. V optike a teda aj v holografii je v praxi významné použitie difrakcia na sústave štrbín. Na obrázku 2.3 je zobrazený prípad dvojice štrbín. Na tienidle sa zobrazí interferenčný vzor, t. j. rôzne široké svetlé a tmavé pružky. Tento obrazec vzniká interferenciou vln v bodoch na tienidle, napr. aj v bode A . Svetlé časti, čiže časti s maximálnou intenzitou sú dôsledkom konštruktívnej interferencie. Naopak, tmavé, s minimálnou intenzitou, sú dôsledkom deštruktívnej interferencie. To, či v bode A vznikne interferenčné maximum alebo minimum, závisí od fázového rozdielu vln l a l_1 , resp. ich dráhového rozdielu.

Podmienkou je, že vzdialenosť d musí byť oveľa väčšia ako vzdialenosť b medzi štrbinami. Na základe tohto kontextu je možné tvrdiť, že uhly α a α_1 sú takmer rovnaké, a preto je možné smer šírenia vln l a l_1 považovať za rovnobežný. Dráhový rozdiel je určený iba vzdialenosťou Δl [8].

Na základe vzniknutého trojuholníka na obrázku 2.3 (tvorený stenami b , Δl a vlnoplocha) môžeme definovať vzťah pre výpočet uhla α a potom aj dráhového rozdielu Δl :

$$\begin{aligned} \sin\alpha &= \frac{\Delta l}{b} \\ \Delta l &= b \sin\alpha \end{aligned} \quad (2.11)$$

Pre interferenčné maximum koherentnej vlny môžeme tvrdiť, že vzniká v mieste stretu s vlnou rovnakej fázy, čiže dráhový rozdiel je celočíselným násobkom vlnovej dĺžky:

$$\Delta l = m\lambda \quad (2.12)$$

Naopak, minimum vznikne, keď sa stretnú vlny s opačnou fázou:

$$\Delta l = (2m + 1) \frac{\lambda}{2} \quad (2.13)$$

V uvedených vzťahoch figuruje m ; je to tzv. rád interferenčného maxima (minima) a platí, že $m \in \mathbb{Z}$. Na základe rovníc 2.11 a 2.12 je určený vzťah pre m -té difrakčné maximum:

$$\sin\alpha = \frac{m\lambda}{b}, \quad (2.14)$$

kde $\sin\alpha$ predstavuje odklon od pôvodného smeru, m rád difrakčného maxima, λ vlnovú dĺžku a b vzdialenosť štrbín. Pre difrakčné minimum by bol vzťah analogický; na základe rovníc 2.11 a 2.13. Z praktických dôvodov sa často používa optická mriežka, ktorá sa skladá zo sústavy rovnako vzdialených štrbín. V tomto prípade by symbol b reprezentoval takzvanú mriežkovú konštantu [8, 4].

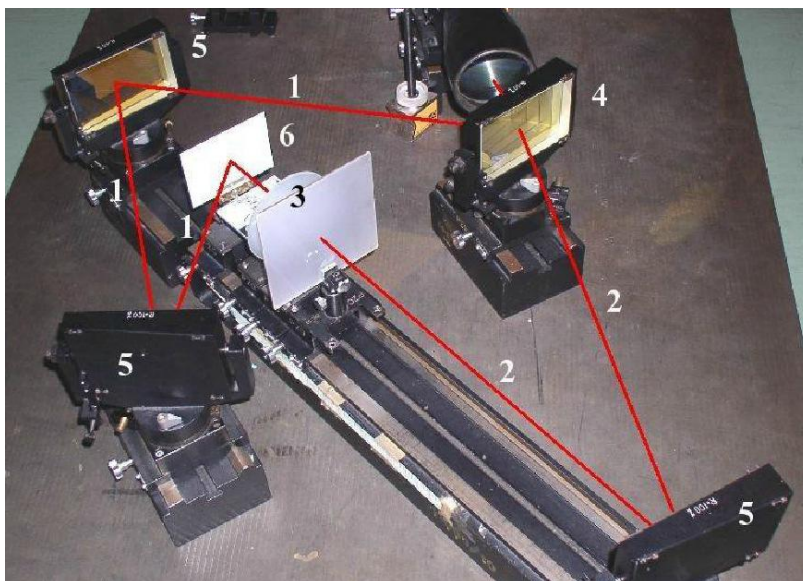
2.2 Optická holografia

V úvode kapitoly bolo zadefinované, že pre túto prácu je relevantná časť holografie, ktorá sa nazýva trojrozmerné zobrazovanie. Tú je ďalej možné rozdeliť na optickú a digitálnu. V tejto časti budú popísané špecifiká prvej spomínanej, konkrétne záznam, rekonštrukcia a niektoré typy hologramov.

Optická holografia sa zaoberá záznamom a rekonštrukciou hologramov s využitím skutočného optického vybavenia. Využíva lasery a záznamové materiály. Základom sú fyzikálne javy a princípy popísané vyššie, ktoré sú aj prakticky využívané pri tvorbe hologramov.

Záznam a rekonštrukcia

Zaznamenanie hologramu sa vykonáva pomocou optickej sústavy. Typicky je potrebných niekoľko zrkadiel, delič – napríklad polopriepustné zrkadlo, záznamové médium, tmavé a stabilné okolité prostredie, zdroj koherentného svetla – laser a samozrejme, samotný objekt, ktorý sa zaznamenáva. Príklad takejto sústavy je na obrázku 2.4.

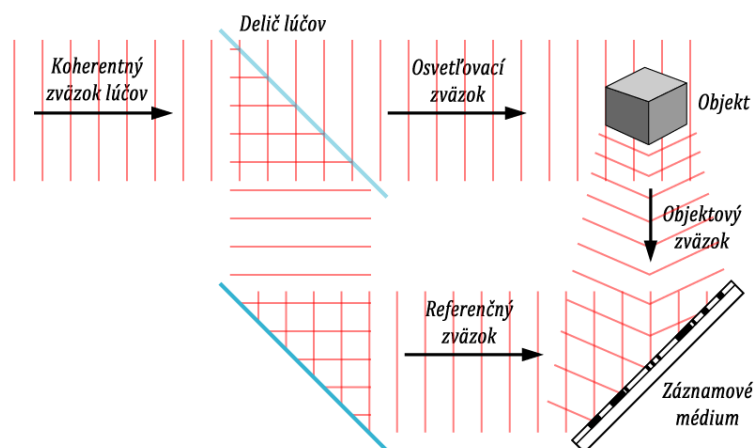


Obrázok 2.4: Záznam hologramu (in-line) pomocou optickej sústavy 1(2) – referenčná (objektová) vlna, 3 – objekt, 4 – delič zväzku, 5 – zrkadlá, 6 – záznamové médium. Obrázok prevzatý z [5].

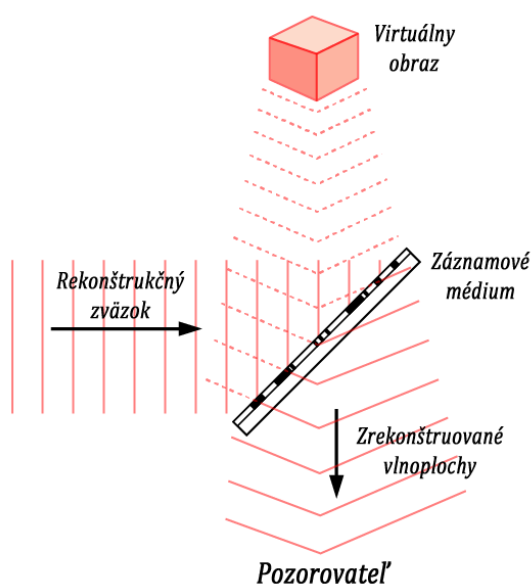
Pre vysvetlenie priebehu záznamu hologramu je využitá jednoduchá schéma z obrázku 2.5. Laser produkuje zväzok koherentných lúčov. Zväzok musí mať dostatočnú šírku aby zaznamenal celý objekt. Následne tento zväzok delič lúčov rozdelí – k úspešnému záznamu hologramu je nutné použiť dva zväzky svetla, referenčný a objektový. Dôvod je jednoduchý. Samotnej jednej vlne nie je možné určiť fázu, hoci na záznam hologramu sa bez informácie o fáze nedá zaobiť. Pri dvojici interferujúcich koherentných vln je to iné, intenzita v každom mieste je určená rozdielom fáz, čím vznikne interferenčný vzor, ktorý sa už zaznamenať dá [5].

Interferenčný vzor, inými slovami hologram, sa skladá z husto usporiadaných prúžkov, podobne ako pri difrakčnej mriežke a pri rekonštrukcii spôsobuje difrakciu svetla žiadaným smerom. Predstavuje zakódovanie scény, t.j. spôsob interferovania svetla zo scény (objektový zväzok) so svetlom zo zdroja (referenčný zväzok).

Zaznamenáva sa na fotosenzitívny materiál, rovnako ako pri fotografii. Typicky ide o rovnaké médium, ale pri holografii je jeho štruktúra veľmi jemná, pretože je schopné zaznamenať oveľa väčšie rozlíšenie.



Obrázok 2.5: Schéma záznamu hologramu.⁴



Obrázok 2.6: Schéma rekonštrukcie hologramu.⁵

Pri rekonštrukcii (obrázok 2.6) je médium, na ktorom je zaznamenaný interferenčný vzor, osvetlené referenčným svetlom pod vhodným uhlom, často pod rovnakým ako pri zázname. Nastáva difrakcia svetla na interferenčnom vzore a dôjde k rekonštrukcii objektovej vlny, resp. k vytvoreniu identického svetelného poľa ako pri zázname, čo nám umožní vidieť objekt [5].

⁴ Obrázok prevzatý z: <http://en.wikipedia.org/wiki/File:Holograph-record.svg>, texty v obrázku sú preložené z anglického originálu.

⁵ Obrázok prevzatý z: <http://en.wikipedia.org/wiki/File:Holography-reconstruct.svg>, texty v obrázku sú preložené z anglického originálu.

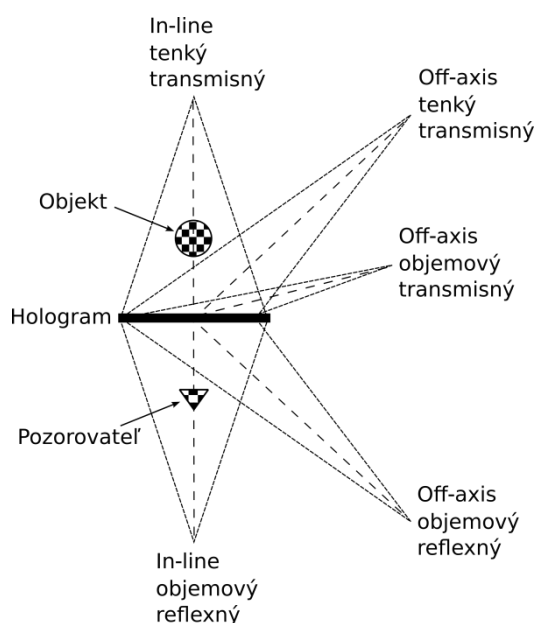
Transmisný a reflexný hologram

Transmisný hologram bol prvýkrát vyrobený v roku 1962 na Michiganskej univerzite. Vytvorili ho vedci Emmett Leith a Juris Upatnieks pri výskume zdokonalenia radaru. Hologram bol zároveň typu „off-axis“, ktorý bude popísaný v ďalšej časti.

Základnou charakteristikou transmisného hologramu je to, že objektový a referenčný zväzok lúčov dopadajú na záznamové médium z tej istej strany. Rekonštrukcia musí prebiehať zväzkom s rovnakými vlastnosťami ako pri zázname, ktorý dopadá pod identickým uhlom. Pozorovateľ sleduje hologram z opačnej strany ako dopadali lúče. Výsledný virtuálny obraz môže byť veľmi ostrý a mať veľkú hĺbku [3].

Špeciálnym prípadom je tzv. „dúhový“ hologram. Ten umožňuje rekonštruovať hologram aj pri bielom svetle. Nevýhodou je eliminácia vertikálnej paralaxy, t. j. pri zmene pohľadu smerom nahor alebo nadol obraz nie je viditeľný.

Reflexný hologram je v určitom zmysle opakom transmisného. Jedná sa o hologram, ktorý je viditeľný pri bielom svetle. Hlavným rozdielom oproti transmisnému je konfigurácia pri zázname – objektová a referenčná vlna dopadajú na opačné strany. Zaznamenaný objekt sleduje pozorovateľ z rovnakej strany ako dopadá rekonštrukčný zväzok lúčov, t. j. strany, na ktorú dopadal referenčný zväzok pri zázname [3, 2].



Obrázok 2.7: Typy hologramov a možné uhly referenčnej vlny.⁶

Pokiaľ ide o reflexný objemový hologram, môže dôjsť k stopercentnej difrakcii odrazeného svetla. V tom prípade bude rekonštrukcia veľmi účinná a kvalita vzniknutého obrazu tak vysoká, že ho bude takmer nemožné rozoznať od skutočného objektu, ktorý sme zaznamenali.

Objemový hologram, niekedy nazývaný aj „tlstý“ je taký hologram, ktorého hrúbka je oveľa väčšia než vlnová dĺžka lúčov svetla zo zdroja pri zázname. Svetlo musí mať správnu vlnovú dĺžku a svetelná vlna správny tvar. Opakom „tlstého“ hologramu je tzv. „thin“, „tenký“ hologram. V tomto

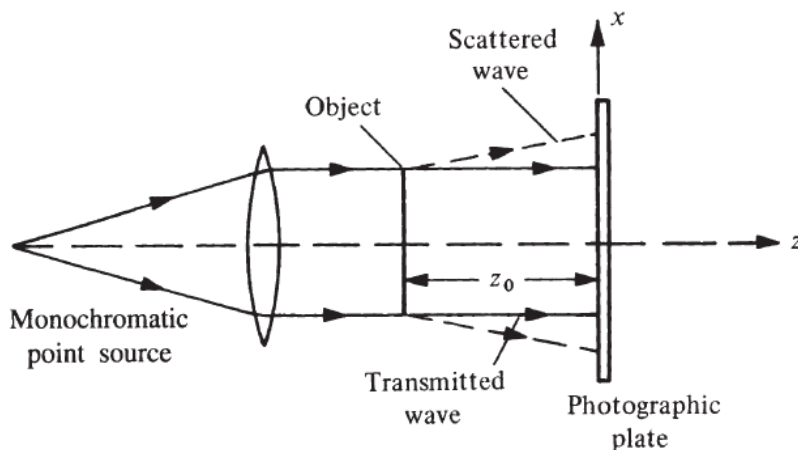
⁶ Obrázok mierne upravený, originál prevzatý z: <http://www.holo.com/holo/book/book5.html>

prípade je hrúbka nahrávacieho média podstatne menšia ako vlnová dĺžka svetla zo zdroja, resp. vzdialenosť medzi prúžkami interferenčného vzoru je oveľa väčšia ako hrúbka záznamového nosiča. Pre podrobnejšie informácie o uvedených typoch hologramov vid' [2].

In-line hologram

Najjednoduchším a najstarším typom hologramu je in-line hologram. Práve ten, ktorý zostrojil Denis Gabor, bol tohto typu. Za svoju jednoduchosť ale platí slabou kvalitou výsledku. Názov je odvodený od rozmiestnenia optickej sústavy pri jeho tvorbe. Zdroj svetla, objekt aj záznamové médium ležia na jednej priamke. Z toho vyplýva, že objekt musí byť do veľkej miery priesvitný, pretože referenčná vlna ním musí prechádzať a v ideálnom prípade aj rovinný [7].

Rozostavenie je uvedené na obrázku 2.8. Na záznamové médium dopadá dvojité svetlo. Prvým je svetlo prechádzajúce objektom reprezentované rovinnou vlnou zo zdroja, u ktorej sa amplitúda ani fáza v rámci záznamovej roviny nemení. Reálna konštanta r označuje jeho komplexnú amplitúdu. Druhým je slabé, objektom rozptýlené svetlo. Zodpovedajúca komplexná amplitúda v každom bode roviny záznamu (x, y) sa bude označovať ako $\tilde{o}(x, y)$, pričom platí, že $|\tilde{o}(x, y)| \ll r$ [2].



Obrázok 2.8: Záznam in-line hologramu.⁷

Výsledná komplexná amplitúda je súčtom vyššie spomenutých zložiek. Intenzita v uvažovanom bode je určená nasledovne:

$$\begin{aligned} I(x, y) &= |r + \tilde{o}(x, y)|^2 \\ &= r^2 + |\tilde{o}(x, y)|^2 + r\tilde{o}(x, y) + r\tilde{o}^*(x, y) \end{aligned} \quad (2.15)$$

Priepustnosť záznamového média sa značí t . Je možné ju zapísať ako:

$$t = t_0 + \beta TI, \quad (2.16)$$

⁷ Obrázok prevzatý z [2].

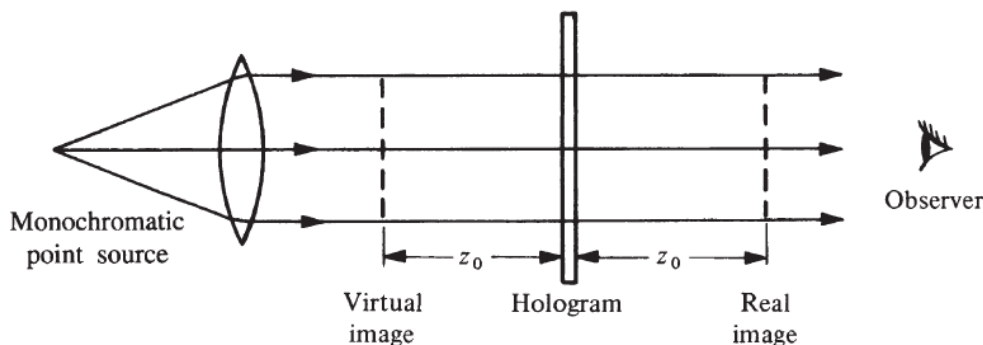
kde t_0 predstavuje konštantnú priepustnosť pozadia, T čas expozície a β je parameter určený fotosenzitívnym záznamovým materiálom. Dosadením intenzity zo vzorca 2.16 vyjde vzťah:

$$t = t_0 + \beta T[r^2 + |\tilde{o}(x, y)|^2 + r\tilde{o}(x, y) + r\tilde{o}^*(x, y)] \quad (2.17)$$

Následne sa dostávame k rekonštrukcii. Hologram je osvetlený rovnakým monochromatickým svetlom ako pri zázname. Komplexná amplitúda prepúšťaná hologramom je potom určená vzťahom:

$$\begin{aligned} \tilde{u}(x, y) &= rt \\ &= r(t_0 + \beta Tr^2) + \beta Tr|\tilde{o}(x, y)|^2 \\ &\quad + \beta Tr^2\tilde{o}(x, y) + \beta Tr^2\tilde{o}^*(x, y) \end{aligned} \quad (2.18)$$

Uvedená rovnica vo výslednom tvare obsahuje na pravej strane štyri členy. Prvý člen, $r(t_0 + \beta Tr^2)$, reprezentuje priamo prepustený zväzok lúčov. Druhý, $\beta Tr|\tilde{o}(x, y)|^2$, je v porovnaní s ostatnými veľmi malý a môžeme ho zanedbať. Tretí člen, $\beta Tr^2\tilde{o}(x, y)$, predstavuje virtuálny obraz a štvrtý, $\beta Tr^2\tilde{o}^*(x, y)$, zase obraz reálny [2].



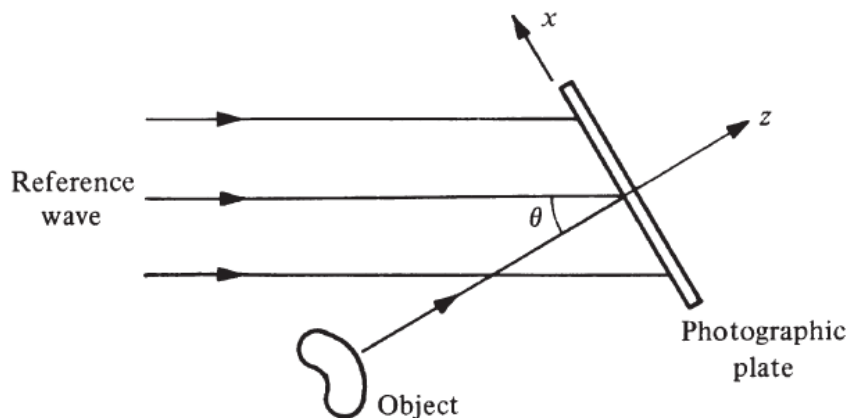
Obrázok 2.9: Rekonštrukcia in-line hologramu.⁸

Rekonštrukcia je znázornená na obrázku 2.9. Všeobecne nízku kvalitu tohto typu hologramu je možné odôvodniť tým, že pre pozorovateľa sa virtuálny a reálny obraz do určitej miery prekrývajú, pričom na pozorovanú oblasť dopadá aj ostré laserové svetlo priamo zo zdroja.

Off-axis hologram

Problém prekrývajúcich sa obrazov rieši off-axis hologram. Tento významný pokrok nastal zmenou v konfigurácii optickej sústavy. Zdrojový lúč sa rozdelí na objektový a referenčný. Referenčný je nasmerovaný priamo na záznamové médium bez akejkoľvek modifikácie. Objektový prechádza scénou a na nosič dopadne rozptýlený objektom, kde interferuje s referenčným a vytvoria hologram. Prakticky sa toto „presmerovanie“ zväzku lúčov dosahuje za použitia sústavy zrkadiel [7]. Jednoduchý náčrt konfigurácie je na obrázku 2.10.

⁸ Obrázok prevzatý z [2].

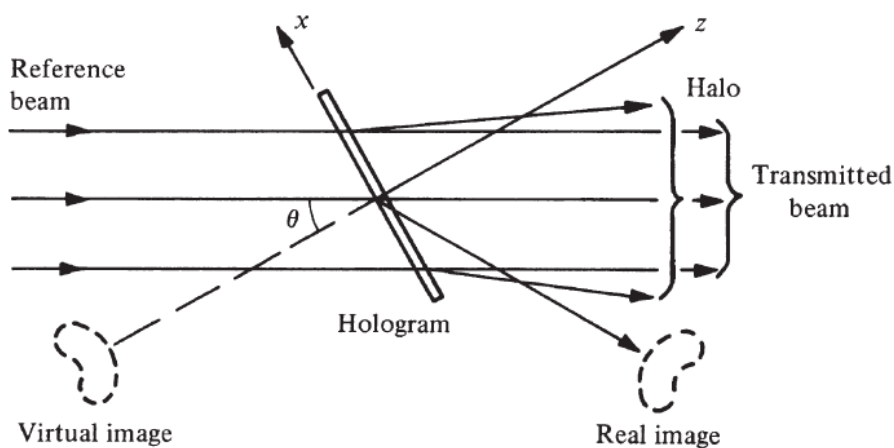


Obrázok 2.10: Záznam off-axis hologramu.⁹

Matematický popis fungovania off-axis hologramu je nasledovný. Komplexná amplitúda referenčnej vlny v každom bode záznamového nosiča je určená ako:

$$\tilde{r}(x, y) = r \exp(i2\pi\xi x), \quad (2.19)$$

kde $\xi = (\sin \theta)/\lambda$, pretože v rovine záznamu sa mení len fáza referenčnej vlny. Veličinu ξ nazývame priestorová frekvencia a udáva počet cyklov na jednotku dĺžky, napríklad 1000 čiar/mm.



Obrázok 2.11: Rekonštrukcia off-axis hologramu.¹⁰

Pre objektovú vlnu sa jej komplexná amplitúda zapisuje ako:

$$\tilde{o}(x, y) = |\tilde{o}(x, y)| \exp[-i\phi(x, y)] \quad (2.20)$$

⁹ Obrázok prevzatý z [2].

¹⁰ Obrázok prevzatý z [2].

Výsledná intenzita bude potom $I(x, y) = |\tilde{r}(x, y) + \tilde{o}(x, y)|^2$. Priepustnosť bola definovaná podľa vzorca 2.16. Po osvetlení hologramu referenčným zväzkom je možné komplexnú amplitúdu prepustenej vlny definovať ako:

$$\begin{aligned}\tilde{u}(x, y) &= \tilde{r}(x, y)t(x, y) \\ &= t'_0 r \exp(i2\pi\xi x) + \beta Tr |o(x, y)|^2 \exp(i2\pi\xi x) \\ &\quad + \beta Tr^2 o(x, y) + \beta Tr^2 o^*(x, y) \exp(i4\pi\xi x)\end{aligned}\tag{2.21}$$

Vzorec 2.21 sa opäť skladá zo štyroch členov. Prvý člen odpovedá priamemu prepustenému svetlu, druhý je zodpovedný za tzv. „halo“ efekt, čiže sprievodné odlesky okolo tohto svetla. Tretí produkuje virtuálny obraz a štvrtý obraz reálny [2].

Ak je uhol odklonu od osy svetla zo zdroja (v náčrte rekonštrukcie na obrázku 2.11 označený symbolom θ) primerane veľký, virtuálny obraz môže byť oddelený od reálneho a taktiež aj od prepúšťaného zväzku lúčov. To, že sa tieto obrazy ani svetlo neprekrývajú, spôsobuje lepšiu kvalitu tohto typu hologramu.

Okrem uvedených typov hologramov existujú ešte ďalšie lišiace sa rôznymi konfiguráciami, napríklad Fourierov hologram alebo tzv. „Image hologram“. Ich charakteristika je nad rámec tejto práce, pre záujemcov viď napríklad [1] alebo [2].

2.3 Digitálna holografia

O digitálnej holografii sa hovorí v dvoch rovinách – názvoslovie nie je doteraz úplne jednotné. V prvej rovine ide o diskretnú „verziu“ optickej holografie, pomocou numerických výpočtov simuluje deje, ktoré fyzicky prebiehajú pri optickej holografii. Často sa používa anglická skratka CGH – „computer generated holography“, počítačom generovaná holografia. V druhom zmysle sa o digitálnej holografii hovorí ako o technológií získavania a spracovania holografických dát, napríklad pomocou CCD kamery alebo obdobného zariadenia, a následného spracovania na počítači [1]. V tejto práci sa pojem digitálna holografia používa v zmysle počítačom generovanej holografie.

Ako bolo v úvode tejto kapitoly načrtnuté, idea tvorby hologramov za pomoci počítačov nie je nová záležitosť. Teoretické základy vznikli prakticky spolu s optickou, myšlienky na využitie výkonu počítačov nasledovali takmer hneď. Ale rovnako ako pri optickej, ktorá čakala na vynález laseru, aj digitálna zastavila svoj vývoj na niekoľko rokov. Zostala v teoretickej rovine, s prísľubom mnohých výhod, až do času, kedy by bol výkon počítačov dostatočný.

Digitálna holografia využíva numerickú simuláciu interakcie svetla so scénou. Simuluje javy popísané v predchádzajúcich kapitolách v prostredí počítača. V uvedenom kontexte sa delí na jednotlivé časti: syntéza hologramu, numerická rekonštrukcia, kompresia, získanie informácií z hologramu či reprodukcia [6]. Táto práca sa bude zaoberať najmä syntézou hologramov, pričom ostatným častiam sa síce nevyhne, ale budú vyriešené za pomoci vhodných nástrojov. Stručný prehľad vybraných metód syntézy bude v nasledujúcej podkapitole.

Metódy syntézy hologramov

Syntéza hologramov je proces numerickej simulácie javov interferencie a difrakcie za účelom vytvorenia hologramu. Uvedené sú niektoré základné postupy.

Prvou metódou je tzv. „diffraction integral evaluation“, metóda počítania difrakčného integrálu. Spočíva v snahe simulovať fyzikálne javy optickej holografie čo najvernejšie. Samotná

scéna je popísaná množinou bodov. Dala by sa považovať za najintuitívnejšiu a najjednoduchšiu metódu. Algoritmus 2.1 sumarizuje jej princíp.

```
foreach  $u(x,y) \in U$  do
     $u(x,y) := 0 + 0j$ ;
    foreach  $ps \in S$  do
         $u(x,y) := u(x,y) + \text{SphericalWave}(ps, x, y)$ ;
    end for
end for
```

Algoritmus 2.1: Metóda syntézy hologramu – „diffraction integral evaluation“.

Princíp popísaný algoritmom 2.1 je relatívne jednoduchý. Pre každý bod optického poľa vypočíta sumu príspevkov ku komplexnej amplitúde zo všetkých bodových zdrojov. Symbol S reprezentuje množinu bodových zdrojov, ps konkrétny bodový zdroj, U predstavuje optické pole [7].

Druhou metódou je „ray-casting“ – metóda vrhania lúčov. Základný princíp spočíva v tom, že medzi dvoma bodmi je vyslaný lúč. Ak nastane prienik medzi lúčom a iným objektom scény, body sa navzájom „nevidia“. Existujú dve varianty tejto metódy.

Prvá prechádza body, z ktorých sa skladá scéna a vysiela z každého z nich lúč smerom ku každému bodu optického poľa. Problém nastáva opäť pri zložitej scéne, resp. veľkom počte bodov, ktoré ju popisujú. Algoritmus tejto varianty je nasledovný:

```
foreach  $u \in U$  do
    foreach  $s \in S$  do
         $I_{s,u} := \text{Cast}(\text{Ray}(s, u))$ ;
        if  $I_{s,u} = \text{null}$  then
             $u := u + \text{SphericalWave}(s)$ ;
        end if
    end for
end for
```

Algoritmus 2.2: Metóda syntézy hologramu – „ray-casting“.

Druhý variant má opačný prístup. Neobsahuje množinu bodových zdrojov, ale bodový zdroj generuje v mieste priesečníku vyslaného lúča s objektom scény. Pre každý bod optického poľa vyšle konštantný počet lúčov. Výhodou je, že počet lúčov nesúvisí so zložitou scény [7].

Existuje viacero ďalších metód, napríklad Fourierova syntéza hologramu, ktorá sa využíva pri tvorbe Fourierových hologramov, ale aj iné. Ďalšie detaily sú nad rámec práce, pre podrobnosti viď napríklad [1].

Techniky optimalizácie

Holografia generovaná pomocou počítačov, ako bolo spomenuté, simuluje fyzikálne javy. Z pohľadu výpočtov sú fyzikálne simulácie všeobecne náročné, a preto existuje potreba aplikovať optimalizácie a rôznymi spôsobmi akcelerovať generovanie hologramov. V tejto časti je stručný prehľad a vymenovanie niektorých postupov. Metóda relevantná pre túto prácu je popísaná detailnejšie.

V predchádzajúcej sekcii v oboch algoritmoch figuruje výpočet komplexnej amplitúdy sférickej vlny (rovnica 2.7). Najnáročnejšia operácia, ktorú obsahuje je výpočet odmocniny pri

výpočte vzdialenosti bodu optického poľa od bodového zdroja, vid' vzorec 4.1. Zložitým je aj výpočet funkcie $f(x) = \exp(x)$ [6]. Viacero úvah sa uberať smerom urýchliť tieto operácie. Uvedený výpočet odmocniny môže byť napríklad aproximovaný pomocou kvadratickej funkcie. Podobný myšlienkový základ má aj metóda využívajúca lineárnu interpoláciu. Účelom je aproximácia vzdialeností pre skupinu bodov hologramu ak sú známe dve východiskové hodnoty. Konkrétny postup je nasledovný.

Optické pole predstavuje časť roviny xy , nech má tvar štvorca so stredom v počiatku súradnicovej sústavy. Pomyselné sa rozdelí na menšie štvorce vhodnej veľkosti, označenej symbolom D . Ďalej existuje jeden bodový zdroj svetla P . Situácia je znázornená na obrázku 2.12.

V rámci zvoleného štvorca sa vypočítajú hodnoty l_1 , l_2 a l_3 , ktoré predstavujú vzdialenosti v troch rôznych bodoch štvorca (vid' obrázok 2.12). Lineárna interpolácia je využitá na zistenie približných hodnôt funkcie $f(x, y)$, ak sú známe dve hodnoty tejto funkcie; funkciu predstavuje výpočet vzdialenosti. Nech má bod v ľavom hornom rohu štvorca, vzdialený od P o vzdialenosť l_1 , súradnice $[x_0, y_0]$, potom $f(x_0, y_0) = l_1$. Pre bod na opačnej, pravej strane štvorca, so súradnicami $[x_0 + m, y_0]$ platí, že $f(x_0 + m, y_0) = l_2$. Analogicky, pre ľavý dolný bod $[x_0, y_0 + n]$, $f(x_0, y_0 + n) = l_3$. Použitím lineárnej interpolácie pre funkciu $f(x, y)$ je možné určiť prírastok v smere osi x aj y , značené Δx a Δy . Platí, že $\Delta x = (l_2 - l_1)/D$ a $\Delta y = (l_3 - l_1)/D$.

Funkciu $f(x, y)$ pre ľubovoľný bod štvorca $[x_m, y_n]$ je možné vyjadriť ako:

$$f(x_m, y_n) = f(x_0 + m, y_0 + n) = f(x_0, y_0) + m \Delta x + n \Delta y \quad (2.22)$$

Uvedené vzťahy sa vo výsledku premietnu nasledovne. Príspevok do optického poľa v bode $[x_0, y_0]$ bude:

$$\tilde{o}(x_0, y_0) = \tilde{A} \exp(jkl_1) \quad (2.23)$$

Pre iný ľubovoľný bod $[x_m, y_n]$ vo štvorci platí:

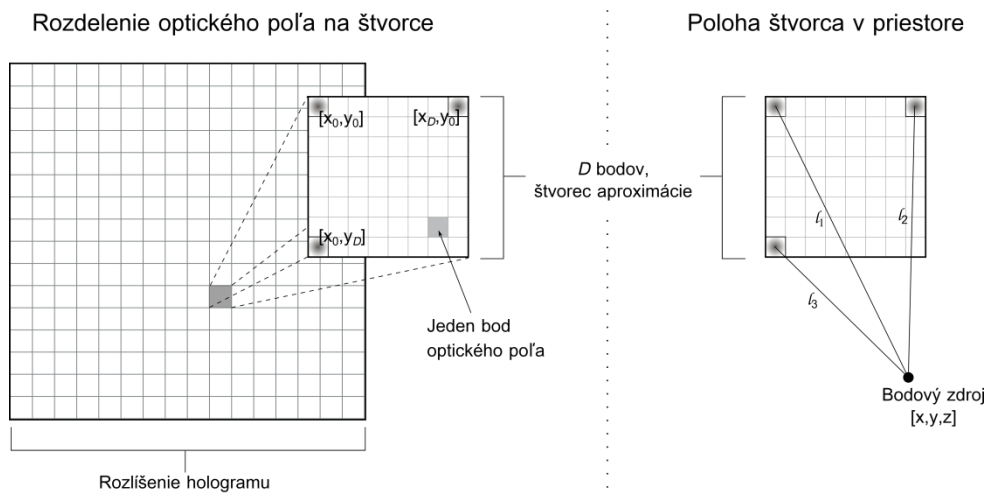
$$\begin{aligned} \tilde{o}(x_m, y_n) &= \tilde{o}(x_0 + m, y_0 + n) = \tilde{A} \exp(jkl_{m,n}) = \tilde{A} \exp[jkf(x_m, y_n)] \\ &= \tilde{A} \exp\{jk[f(x_0, y_0) + m \Delta x + n \Delta y]\} \\ &= \tilde{A} \exp(jkl_1) \exp(jkm \Delta x) \exp(jkn \Delta y) \\ &= \tilde{o}(x_0, y_0) \exp(jkm \Delta x) \exp(jkn \Delta y) \end{aligned} \quad (2.24)$$

Rozdiel príspevkov v dvoch susedných bodoch - v smere osi x - optického poľa je potom:

$$\tilde{o}(x_m + 1, y_n) = \tilde{o}(x_m, y_n) \exp(jk\Delta x) \quad (2.25)$$

Člen $\exp(jk\Delta x)$ je v rámci jedného štvorca konštantný a preto ho je možné vypočítať dopredu. Toto platí samozrejme analogicky aj pre os y . Uvedeným spôsobom je vlastne pri väčšine výpočtov príspevku optického poľa nahradený výpočet obsahujúci trojnásobné odčítanie a druhú mocninu, dve sčítania, odmocninu, násobenie a výpočet komplexnej funkcie $\exp(x)$ za jednu operáciu komplexného násobenia [19]. Dôležitou časťou je určenie vhodnej veľkosti štvorca. Konkrétne zlepšenie rýchlosti je uvedené v časti Testovanie a výsledky.

Okrem spomenutej optimalizácie, ktorá bola z dôvodu relevantnosti pre túto prácu popísaná obšírnejšie, existuje celá rada ďalších zjednodušení a optimalizácií, napríklad výpočet hologramu priamo, bez počítania optického poľa a následného pridávania referenčnej vlny, ďalej výpočet diskretnej podoby difrakčného vzorca Rayleigh-Sommerfeld, Fresnelova aproximácia, použitie dopredu spočítaných tabuliek hodnôt sínusov, kosínusov a vzdialeností, spoločný výpočet optického



Obrázok 2.12: Rozdelenie optického poľa na štvorce pri optimalizácii počítania vzdialeností.

poľa pre priamku – ak množstvo bodov scény leží na tej istej priamke, zhlukovanie bodových zdrojov do trojuholníkov či vykresľovanie cez spektrum [6].

3 Techniky a aplikácie

V tejto časti budú všeobecne popísané viaceré techniky a riešenia využiteľné pri počítačom generovanej holografii. Opäť, ako bolo spomenuté v úvode kapitoly 2, nejde o encyklopedický prehľad či vymenovanie všetkých možných techník a aplikácií využiteľných v digitálnej holografii; ide o popis záležitostí relevantných pre túto prácu, a to najmä o sadu nástrojov pre digitálnu holografiu SDK Holography Toolkit a jej jednotlivé časti, ďalej rozhranie OpenMP pre programovanie paralelných aplikácií či programovací model C++ AMP. Vysvetlené budú aj pojmy „wireframe“ model a „point cloud“.

3.1 SDK Holography Toolkit

SDK Holography Toolkit (ďalej len HoloToolkit) je súbor nástrojov pre digitálnu holografiiu, ktorá sa skladá z niekoľkých aplikácií a rôznych podporných súčastí, napríklad skriptov pre program Matlab, špecifikácie formátov, konfiguračných súborov či knižníc. Vyvinutý bol na Západočeskej univerzite v Plzni. Nasleduje stručný popis častí najvýznamnejších z pohľadu tejto práce. Pre podrobný popis viď manuál HoloToolkitu [10].

Prvou je aplikácia **DFToHologram**. Tá vykonáva výpočet hologramu z optického poľa uloženého vo formáte DF. Výpočet spočíva v pridaní rovinnnej referenčnej vlny k objektovej (optické pole predstavuje objektovú vlnu). Následne prebehne výpočet intenzity. Výstupné formáty sú rôzne, ale všeobecne ide o obrázok, ktorý je v odtieňoch šedej či binárnej forme, napríklad vo formáte PNG.

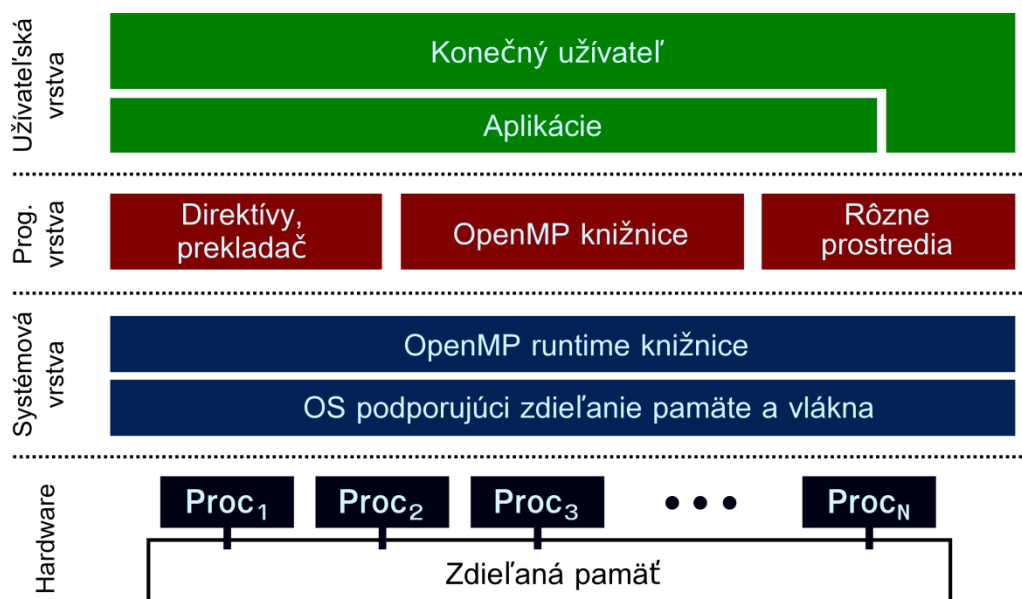
HoloPropagLarge je nástroj na propagáciu optického poľa medzi dvoma paralelnými rovinami. Na propagáciu využíva algoritmus „angular spectrum“. Proces propagácie je riadený vlastným interpretovaným jazykom. Vstupom môže byť súbor DF či PNG obrázok, výstupom môžu byť tak isto DF alebo PNG súbor, prípadne TGA. Zjednodušene by sa dalo povedať, že tento program vizualizuje výsledok numerickej rekonštrukcie hologramu.

Formát DF, ktorý je spomínaný vyššie, špecifikuje súbor, ktorý obsahuje optické pole a ostatné potrebné informácie o ňom, t. j. rozlíšenie, veľkosť vzorky optického poľa, vlnovú dĺžku. Optické pole je dvojrozmerné pole vzoriek – komplexných čísel. Vo formáte DF je uložené štýlom „riadok za riadkom“.

3.2 OpenMP

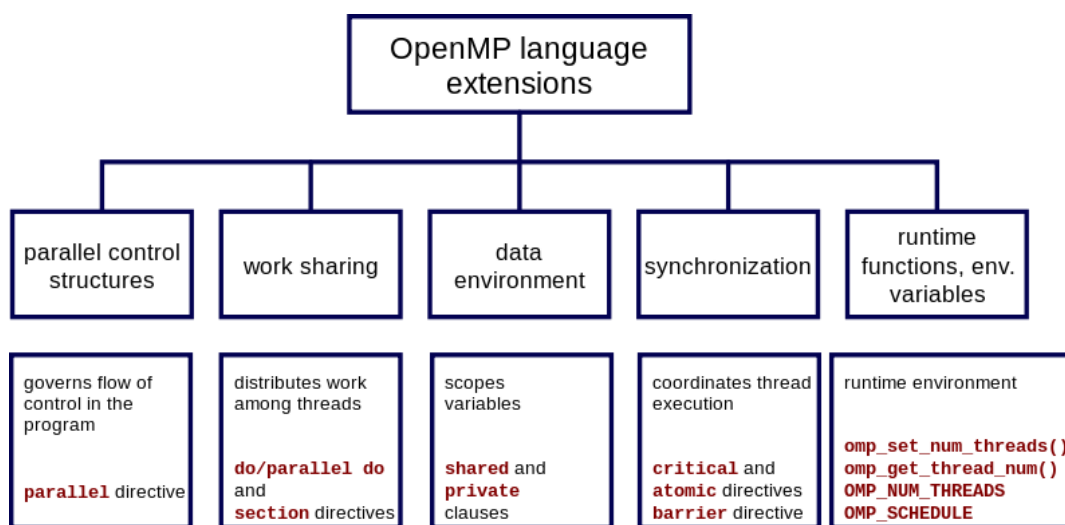
OpenMP je rozhranie pre programovanie aplikácií (API), ktorého význam spočíva v myšlienke zabezpečiť jednoduché paralelné programovanie s využitím zdieľanej pamäte. V súčasnosti sa dá o OpenMP hovoriť ako o štandarde pre paralelizáciu aplikácií, hoci ide skôr o dohodu medzi členmi ARB (Architecture Review Board) – konzorciom najvýznamnejších predajcov hardvéru a softvéru, členmi sú napr. IBM, AMD, Intel, HP, NVIDIA, NEC, Texas Instruments, a. i.

OpenMP je výsledkom snahy o prenositeľné, užívateľsky prívetivé, škálovateľné a efektívne paralelné programovanie. Avšak nejde o nový programovací jazyk – jednotlivé záznamy môžu byť zapísané v rámci sekvenčného programu v niektorom z programovacích jazykov C, C++ alebo Fortran. Pomocou týchto záznamov je možné definovať rozdelenie požadovanej práce medzi vlákna, ktoré budú vykonávané na rozdielnych procesoroch počítača aj prístup k pamäti [14].



Obrázok 3.1: OpenMP schéma riešenia.¹¹

Spomínané záznamy sú v skutočnosti direktívy prekladača. Užívateľ tak prekladaču oznámi, ktoré inštrukcie majú byť vykonané v paralelnom režime a ako ich prerozdeliť medzi vlákna, ktoré budú bežať. OpenMP direktíva prekladača je inštrukcia v špeciálnom formáte a na jej využitie je potrebný prekladač s podporou OpenMP. Významnou výhodou je fakt, že ak prekladač nepodporuje OpenMP, direktívy sa javia ako komentáre, preskočí ich a program preloží štandardným spôsobom.



Obrázok 3.2: Základné časti OpenMP.¹²

Samotné API poskytuje len niekoľko direktív, napriek tomu sú postačujúce na množstvo potrieb pri paralelizácii. Z pohľadu programátora je dôležité identifikovať miesto možnej paralelizácie a direktívy vhodne použiť. V prípade jazykov C/C++ sa využíva direktíva `#pragma` [13].

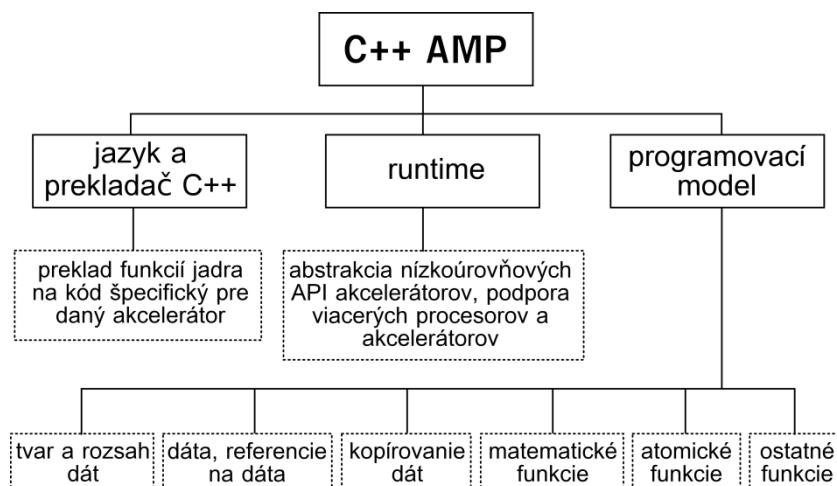
¹¹ Obrázok upravený a prevzatý z:

<http://cs.wikipedia.org/wiki/OpenMP#mediaviewer/Soubor:OMPSchema2.gif>.

¹² Obrázok prevzatý z: http://en.wikipedia.org/wiki/File:OpenMP_language_extensions.svg.

3.3 C++ AMP

C++ Accelerated Massive Parallelism (AMP) predstavuje technológiu firmy Microsoft určenú na akceleráciu C++ aplikácií tým, že im umožní beh na dátovo paralelnom hardvéri, ktorý je v súčasnosti reprezentovaný najmä grafickými procesormi (GPU). V prípade neprítomnosti vhodného GPU prebehnú operácie štandardne na procesore CPU. Implementácia je postavená nad DirectX, čím je umožnená prenositeľnosť naprieč rôznymi hardvérovými platformami a nezávislosť od výrobcu. Špecifikácia zavádza na popis dátovo paralelného hardvéru pojem akcelerátor (z angl. *accelerator*) [12].



Obrázok 3.3: Konceptia C++ AMP.¹³

Programovací model C++ AMP dáva vývojárovi plnú kontrolu nad prístupom k akcelerátoru. Môže explicitne riadiť všetku komunikáciu medzi procesorom (CPU) a akcelerátorom, synchronnú aj asynchronnú. Operácie na akcelerátore sú vyjadrené na vysokej úrovni abstrakcie – využívajú viacrozmerné polia, vysokoúrovňové funkcie pre prácu s nimi či viacrozmerné operácie indexovania. Založené sú na rozsiahlej podmnožine jazyka C++ [11].

V súčasnosti je C++ AMP implementované len v systémoch Windows, hoci celkovo má rozhranie otvorenú špecifikáciu a následná implementácia na iných platformách je veľmi pravdepodobná.

Základom je použitie výrazu `restrict(amp)`, ktorý môže byť použitý na ľubovoľnú funkciu. Pomocou neho je deklarované, že funkcia môže byť vykonaná na príslušnom akcelerátore. Kľúčové slovo `restrict` oznamuje prekladaču, že je potrebné skontrolovať, či funkciu možno akcelerovať a jej telo neobsahuje nepovolené príkazy či definície.

C++ AMP je dostupné pomocou hlavičkového súboru `<amp.h>` v mennom priestore `concurrency`. Kľúčové triedy, ktoré je možné použiť sú:

- `array` – kontajner pre dáta na akcelerátore
- `array_view` – obalenie pre dáta
- `index` – N-rozmerný bod
- `extent` – N-rozmerná veľkosť

¹³ Podklady k obrázku sú popísané v [11].

Samotná akcelerácia prebieha pomocou konštrukcie `parallel_for_each`. Tá vyjadruje paralelný cyklus typu `for` - v tele obsahuje kód, ktorý je možné urýchliť pomocou zvoleného akcelerátora [12].

3.4 „Wireframe“ a „point cloud“

V počítačovej grafike predstavuje **„wireframe“ model** vizuálnu reprezentáciu objektu bez povrchu. Ide o takzvaný „drôtený“ model, pretože reprezentuje len hrany – rovnako by vyzeral model objektu vytvorený len z drôtu. Z tohto dôvodu je vhodný pri zložitých výpočtoch, jeho renderovanie a prípadné transformácie, napríklad rotácia alebo zmenšenie, je relatívne nenáročné.

Wireframe model, respektíve scéna s viacerými modelmi, je všeobecne definovaná dvomi tabuľkami. Prvou je tabuľka vrcholov, v ktorej sú špecifikované súradnice vrcholov modelu. Druhou je tabuľka hrán, v ktorej je špecifikované, ktoré vrcholy sú spojené s ktorými, čím je vlastne udaná hrana [15][16].

„Point cloud“ je množina bodov určitého súradnicového systému. Zvyčajne – inak tomu nie je ani v tejto práci – ide o trojrozmerný súradnicový systém, kde je každý bod definovaný trojicou súradníc $[x, y, z]$. Bod môže obsahovať viac úrovni informácií, napríklad osvetlenie, farba a. i. Point cloud reprezentuje vonkajší povrch objektu [18].

Uvedené pojmy pochádzajú z anglického jazyka a hoci by bolo možné nájsť slovenský (český) ekvivalent - drôtený model, resp. mračno bodov - v literatúre sa uvádzajú takmer výlučne v pôvodnom tvare. V ďalšom texte preto budú výrazy wireframe a point cloud používané bez úvodzoviek v anglickom tvare.

4 Zhodnotenie stavu a plán práce

V tejto kapitole je zhodnotený súčasný stav na poli jednoduchých aplikácií zaoberajúcich sa digitálnou holografiou. Popísané sú terajšie komplikácie, s ktorými je možné sa stretnúť, či náležitosti, ktoré by sa mali riešiť. Pri čítaní zhodnotenia treba mať na mysli, že je písané z pohľadu tejto práce a v rámci súčasných možností a schopností autora.

Zároveň navrhuje obecný postup riešenia a stáva sa tak detailným zadáním práce. Obsahuje dekompozíciu riešenia na jednotlivé časti, stanovuje parametre riešenia, akým smerom by bolo vhodné sa uberať a ako bude možné zhodnotiť výsledky.

4.1 Zhodnotenie súčasného stavu

Záujem o holografiiu od jej vzniku kolíše – vždy keď sa dosiahne technologický strop v určitej oblasti, záujem prirodzene upadá. Takisto naopak, keď sa prekoná určitá prekážka technologického pokroku, obnoví sa práca a výskum na ďalších nových možnostiach.

V súčasnosti je holografia pomerne populárna vedecká oblasť, napriek tomu existuje len málo aplikácií zaoberajúcich sa holografiou na úrovni, ktorá je niekde medzi výrazne odbornou a úplne základnou. Existuje napríklad program CGHMaker¹⁴ od CorticalCafe, ktorý umožňuje syntézu optického poľa aj s následným vytvorením hologramu, ale nepredpokladá následnú výrobu skutočného hologramu s rozumnými rozmermi. Vytvorený je ako určitá forma technickej hračky, kde si každý bežný používateľ počítača môže vyrobiť a zrekonštruovať hologram za pomoci laserovej tlačiarne a laserového ukazovadla, avšak kvalita výsledku je mizivá. Samotná aplikácia je naprogramovaná v jazyku Java a jej rýchlosť je rozhodne taktiež slabšou stránkou. Pre implementáciu náročných výpočtov, ktoré sú pri digitálnej holografii na vyššej úrovni nevyhnutné, je najvhodnejším bežným prostriedkom programovací jazyk C/C++.

Obecne, tvorba hologramov nie je triviálna záležitosť a vyžaduje si viacero kompromisov. Pri optickej holografii je jednoznačnou výhodou jej kvalita výsledku. Problémy spočívajú v zložitom systéme záznamu a vysokej citlivosti na rôzne drobné výkyvy, napríklad mierny pohyb počas záznamu úplne znehodnotí výsledok. Problémom je aj to, že scéna, ktorá sa zaznamenáva, musí byť reálne prítomná. Toto je najväčšou výhodou digitálnej holografie. Dokáže použiť ľubovoľnú scénu ako podklad pre hologram, avšak za cenu nižšej kvality. Počítač má obmedzenú presnosť a spojené fyzikálne deje musí diskretizovať. V prípade holografie nie je tento proces vôbec jednoduchý. Zložitosť operácií a ich množstvo je značne náročné na zdroje počítača a v súčasnosti predstavuje najväčšiu prekážku v ďalšom vývoji. Optimalizácia a zrýchľovanie výpočtov je preto nevyhnutnosťou. S narastajúcou mierou optimalizácií sa však zvyčajne znižuje kvalita a je potrebné nájsť primeranú rovnováhu medzi časom výpočtov a kvalitou výsledku. Pre lepšiu ilustráciu nárokov digitálnej holografie je uvedený príklad.

Príklad 1: Uvažujme metódu počítania difrakčného integrálu (algoritmus 2.1). Objekt nahradíme množinou bodov s veľkosťou $N_{\text{obj}} = 40\,000$ bodov. Požadované parametre hologramu sú nasledovné: veľkosť jedného bodu hologramu bude $2\,\mu\text{m}$, rozlíšenie $10\,000 \times 10\,000$ bodov, t. j. reálny hologram má rozmery $2\,\text{cm} \times 2\,\text{cm}$, čo je primeraná veľkosť na rekonštrukciu. Potom počet výpočtov $N_{\text{výp}}$ bude:

¹⁴ Dostupný na http://www.corticalcafe.com/prog_CGHmaker.htm

$$N_{výp} = 10\,000 * 10\,000 * 40\,000 = 4\,000\,000\,000\,000,$$

čo sú 4 bilióny výpočtov. Zároveň pri každom jednom behu sa počíta netriviálny výpočet príspevku optického poľa – rovnica sférickej vlny (rovnica 2.7). Tá obsahuje výpočet vzdialenosti bodu optického poľa $P_h = [x_h, y_h, z_h]$ od bodového zdroja $P_{ps} = [x_{ps}, y_{ps}, z_{ps}]$:

$$l = \sqrt{(x_h - x_{ps})^2 * (y_h - y_{ps})^2 * (z_h - z_{ps})^2}, \quad (4.1)$$

Ďalšou náročnou operáciou je aj výpočet funkcie $f(x) = e^x$.

Nezanedbateľnou nie je ani pamäťová náročnosť. Optické pole, ako je spomínané vyššie, je pole komplexných čísel, t. j. skladá sa z reálnej a imaginárnej zložky. Ak by bol na uloženie čísel využitý dátový typ `float`, ktorého veľkosť je 4 bajty, samotné optické pole zaberie v pamäti M bajtov. Pre uvedené rozlíšenie je hodnota M nasledovná:

$$M = 10\,000 * 10\,000 * 2 * 4\,B = 800\,000\,000\,B,$$

čo je približne 763 MB. Je zrejmé, že ako pamäťové tak výpočtové nároky sú enormné a je potrebné tento problém predpokladať a zohľadniť ho v návrhu.

Dôležitý je aj vhodný výber algoritmu. V kapitole 2.3 boli spomenuté dva prístupy výpočtov – „diffraction integral evaluation“ a „ray-casting“. Existujú viaceré ďalšie prístupy, napríklad je možné využitie diskretnej Fourierovej transformácie, kedy je nutné nahradiť objekt za dvojrozmerný obrázok objektu. Trojrozmernosť je síce možné simulovať sériou obrázkov [holografie2013], z pohľadu zadania a povahy scény – trojrozmerný wireframe model prevedený na množinu bodov – je však výhodné túto možnosť vylúčiť a sústrediť sa len na popísané dve.

Na základe charakteristiky metódy výpočtu difrakčného integrálu uvedenej v kapitole 2.3 je možné všimnúť si hlavné nevýhody. Pri veľkom počte bodových zdrojov, ktoré reprezentujú scénu, je algoritmus neefektívny a výpočet by trval veľmi dlho. Ak by sme chceli implementovať aj osvetľovací model a riešenie viditeľnosti v scéne, dve základné požiadavky pri realistickom zobrazovaní, náročnosť by bola ešte oveľa vyššia. Pri pozornom naštudovaní algoritmu je ale zrejma výhoda – veľmi jednoduchá paralelizácia.

Čo sa týka kvality výsledku metódy vrhania lúčov, prekonáva nedostatky integrálnej metódy vrátane spomínaných javov viditeľnosti či osvetľovacieho modelu. Avšak táto jej výhoda má svoju cenu – z dôvodu množstva výpočtov je tiež extrémne pomalá.

4.2 Návrh riešenia aplikácie

Cieľom tejto práce je vytvoriť aplikáciu, ktorá bude implementovať syntézu optického poľa zvolenou metódou, následne ju optimalizovať vybranými postupmi a tie porovnať. Hlavným produktom, okrem aplikácie, by mal byť reálny hologram, ktorý sa vyrobí z vypočítanej predlohy v digitálnej podobe a zrekonštruje v laboratórnych podmienkach. Primárna optimalizácia je popísaná v kapitole 2.3, časť Techniky optimalizácie. Nezanedbateľnou súčasťou bude vytvorenie primeranej scény tak, aby bola vizuálne zaujímavá, ale neprekročila vhodnú mieru jednoduchosti. Následne popíšem plán práce detailnejšie.

Scéna a syntéza optického poľa

Vstupom programu by mali byť parametre o dvoch elementoch – o scéne, ktorej holografický záznam bude vytváraný a informácie nutné k syntéze optického poľa, t. j. vlnová dĺžka zdroja svetla, rozlíšenie hologramu, vzorkovanie, atď. Scéna bude tvorená elementárnym trojrozmerným „wireframe“ objektom, ktorého vybrané steny, prípadne jedna stena, budú obsahovať určitý obrazec. Pridanie obrazca má prostý dôvod - aby scéna vyzerala vizuálne atraktívnejšie.

Pre generovanie optického poľa je vhodné previesť scénu na množinu bodov, ktoré budú pri fyzikálnych výpočtoch predstavovať bodové zdroje svetla. V tejto fáze teda prebehne prevod scény z wireframe modelu na point cloud. Potom nad touto množinou bodových zdrojov prebehne výpočet optického poľa zvolenou metódou.

Pretože riešenie viditeľnosti v prípade scény v tejto práci nie je potrebné, rovnako ako osvetľovacieho modelu, bude najvhodnejšie zamerať sa na metódu počítania difrakčného integrálu a urýchliť ju. Bolo by vhodné spracovať návrh tak, aby bola umožnená neskoršia implementácia pokročilejších metód syntézy, avšak pre túto prácu je zvolená jednoduchšia metóda postačujúca. Vybraná metóda sa okrem toho dá výborne akcelerovať pomocou paralelizácie.

Náročnosť na zdroje a zohľadnenie v návrhu

Riešenie pamäťovej náročnosti nie je komplikované. Keďže nezáleží na poradí výpočtu príspevkov do optického poľa, je možné rozdeliť bodové zdroje do viacerých celkov. Rovnako je možné rozdeliť hologram – ak bude jeho rozlíšenie príliš veľké – na vhodne veľké časti, napr. štvorce alebo niekoľko riadkov. Výpočet potom bude v jeden moment prebiehať len pre jednotlivé časti a v pamäti bude len to, čo je aktuálne potrebné.

Čo sa týka výpočtovej náročnosti, ponúka sa viacero alternatív. Použitie distribuovaných výpočtov, počítanie pomocou číslicových obvodov FPGA, paralelizácia v rámci CPU, či akcelerácia na grafickej karte pripadajú do úvahy. Na základe osobných preferencií a záujmu sa ďalej zameriam na posledné dve možnosti.

Pre paralelizáciu na CPU je výhodné využiť rozhranie OpenMP, ktoré je v súčasnosti obľúbené a užívateľsky prívetivé. V kombinácii s využitím spomínanej optimalizácie by mohlo priniesť zaujímavé zlepšenie. Využitie grafického procesoru GPU na urýchlenie výpočtov sa stáva čoraz populárnejšie. Existuje niekoľko možností; pre grafické karty od výrobcu NVIDIA je možné použiť architektúru CUDA, pre výrobcu AMD je to súbor nástrojov ATI Stream. Od firmy Microsoft pochádza technológia C++ AMP, ktorá umožňuje využitie aj GPU ale je doposiaľ viazaná na operačný systém Windows, aj keď existuje rozumný predpoklad, že v budúcnosti sa to zmení. Najobecnejším riešením je využitie frameworku OpenCL, ktorý umožňuje prístup k rôznym hardvérovým platformám vrátane grafických kariet. Na základe faktu, že implementáciou v OpenCL sa zaoberalo už niekoľko prác, a vôbec zo záujmu o túto technológiu, som sa rozhodol využiť C++ AMP.

Rekonštrukcia a výroba

Samotné vypočítané optické pole predstavuje záznam len objektovej vlny. K nemu je potrebné pripočítať ešte referenčnú vlnu dopadajúcu na rovinu hologramu pod vhodným uhlom. Z tohto výsledného poľa sa potom vypočíta intenzita a výsledok sa uloží ako pole pixelov vo vhodnom formáte. Pre výrobu treba binarizovaný obrázok v bezstratovom formáte. Implementácia týchto dvoch krokov je nad rámec práce a na ich prevedenie bude využitá sada nástrojov HoloToolkit popísaná v kapitole 3.1, pretože presne vyhovuje uvedeným potrebám.

Po overení výsledku numerickou simuláciou sa na základe hologramu v podobe obrázku nechá vyrobiť fyzický hologram, ktorý by bolo možné zrekonštruovať v laboratórnych podmienkach. Na základe úspešnej predchádzajúcej spolupráce sa hologram vyrobí na ZČU¹⁵ a rekonštruovať sa bude v optickom laboratóriu FIT¹⁶.

Výroba hologramov je veľmi špecifická oblasť a existuje niekoľko základných riešení a postupov. Líšia sa vo viacerých parametroch. Stručný súhrn je v tabuľke 4.1. V rámci tejto práce môžeme zúžiť uvažovanie nad vhodnou technológiou na laserovú litografiu alebo osvitovú jednotku, keďže elektrónová litografia je extrémne drahá a aj výpočtová náročnosť by bola z dôvodu jemnejšieho vzorkovania oveľa vyššia. Opačný extrém, použitie laserovej tlačiarne, je pre trojrozmernú scénu úplne nevhodný – difrakčný uhol a kvalita nedosahujú požadované hodnoty.

technológia	veľkosť vzorky	maximálny difrakčný uhol	rýchlosť výroby	cena (rádovo)	kvalita
elektrónová litografia	0,1 μm	90°	1 mm ² /min	~ 10000 Kč/cm ²	excelentná
laserová litografia	1 μm	20°	4 mm ² /min	~ 1000 Kč/cm ²	výborná
osvitová jednotka	10 μm	2°	-	~ 0,20 Kč/cm ²	priemerná
laserová tlačiareň	100 μm	0,5°	-	< 0,02 Kč/cm ²	slabá

Tabuľka 4.1: Prehľad technológií výroby hologramov.¹⁷

Zhrnutie stanovených parametrov a overenie výsledkov

Na základe úvah popísaných vyššie v tejto kapitole pristúpim k zhrnutiu stanovených parametrov a tie, ktoré chýbajú, stanovím.

Aplikácia bude naprogramovaná v jazyku C++. Vstupná scéna a parametre budú špecifikované v popisnom XML súbore. Vlastnosti hologramu sa odvíjajú od možností optického laboratória na FIT, t. j. rozmery hologramu by nemali presiahnuť približne 2 x 2 cm, zdroj svetla je HeNe (hélio-neónový) laser s vlnovou dĺžkou 632,8 nm. Počet bodových zdrojov by sa mal pohybovať rádovo v desiatkach tisícov bodov.

Syntéza optického poľa bude prebiehať metódou počítania difrakčného integrálu s urýchlením pomocou optimalizácie algoritmu a využitím OpenMP (CPU verzia) a C++ AMP (GPU verzia). Výsledky a prínos rôznych implementácií budú porovnané.

Holografia, keďže pracuje rádovo s veľkosťami na úrovni vlnovej dĺžky svetla, je citlivá na chyby – či už pri fyzickej alebo numerickej rekonštrukcii. Testovanie a overenie správnosti preto prebehne pomocou nástrojov HoloToolkitu formou numerickej rekonštrukcie. V prípade úspešných výsledkov sa získaný hologram nechá vyrobiť technológiou laserovej litografie. Záverečné overenie funkčnosti programu a úvah v práci vôbec by tak mohlo prebehnúť vo forme rekonštrukcie vyrobeného hologramu v optickom laboratóriu.

¹⁵ Západočeská univerzita v Plzni. Web: <http://www.zcu.cz/>

¹⁶ Fakulta informačních technologií, VUT. Web: <http://www.fit.vutbr.cz/>

¹⁷ Informácie pochádzajú z [4].

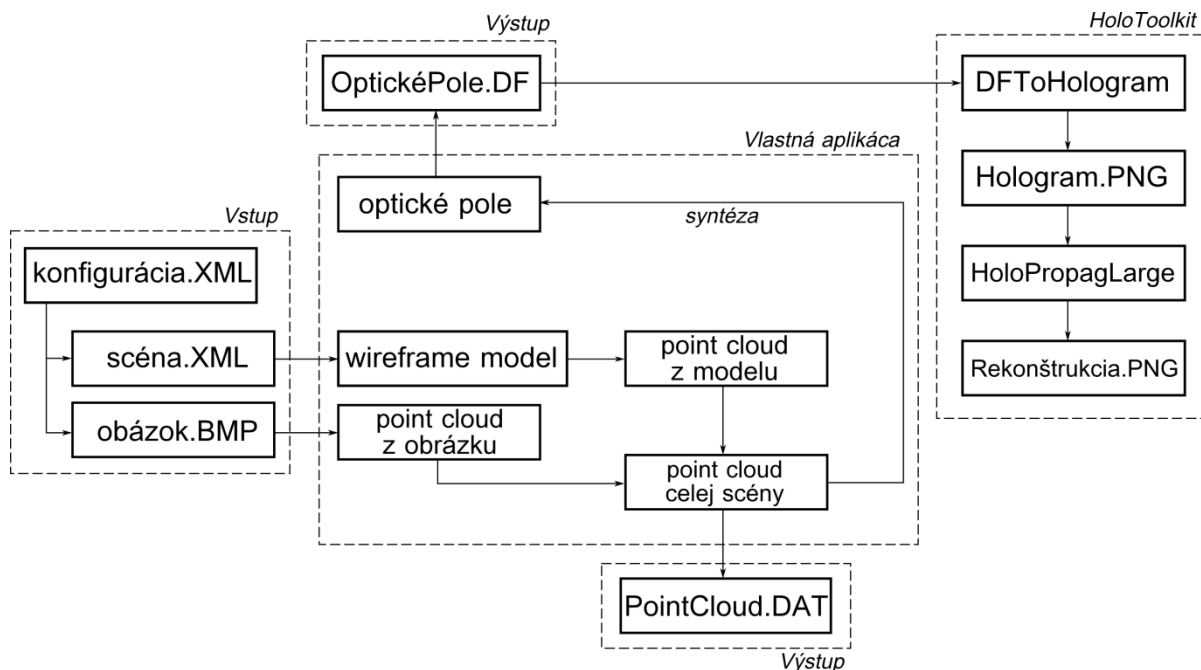
5 Implementácia

Táto kapitola obsahuje popis vlastnej práce, najmä z pohľadu programovej implementácie stanoveného riešenia. Na začiatku je popísaná základná koncepcia a všeobecné náležitosti, nasleduje podrobnejší rozbor niektorých netriviálnych častí. Na záver kapitoly je uvedené testovanie a vyhodnotenie dosiahnutých výsledkov.

5.1 Celková koncepcia aplikácie

Výsledným produktom práce je aplikácia schopná výpočtu optického poľa rôznymi spôsobmi. Použitými implementačnými jazykmi je kombinácia jazykov C a C++, kde je riešenie celkovo usporiadané objektovo (C++), ale využíva množstvo konštrukcií a pomocných knižníc jazyka C. Vývoj prebiehal na platforme Windows 7 v prostredí Visual Studio 2013, ktoré obsahuje množstvo nástrojov na podporu produktivity a vstavaný prekladač MSVC v uvedenej verzii podporuje štandard C++11.

Aplikácia sa spúšťa pomocou príkazového riadku. Vstupom sú tri súbory. Prvým je konfiguračný súbor vo formáte XML, ktorý obsahuje všetky parametre programu a nahrádza ich zadávanie z príkazového riadku, pretože parametrov je značné množstvo. Druhý vstupný súbor je popis scény opäť vo formáte XML. Ten definuje wireframe objekt a počet bodov point cloud-u, ktorý z neho bude vygenerovaný. Tretím súborom je obrázok vo formáte BMP, ktorý je možné umiestniť na stenu objektu.



Obrázok 5.1: Celková koncepcia aplikácie.

Aplikácia vykoná nad vstupmi postupnosť nasledovných činností. Z popisu scény vytvorí internú reprezentáciu wireframe objektu. Z neho vygeneruje point cloud. Vstupný obrázok je tiež prevedený na point cloud, keďže pre výpočet optického poľa je táto reprezentácia v danom kontexte

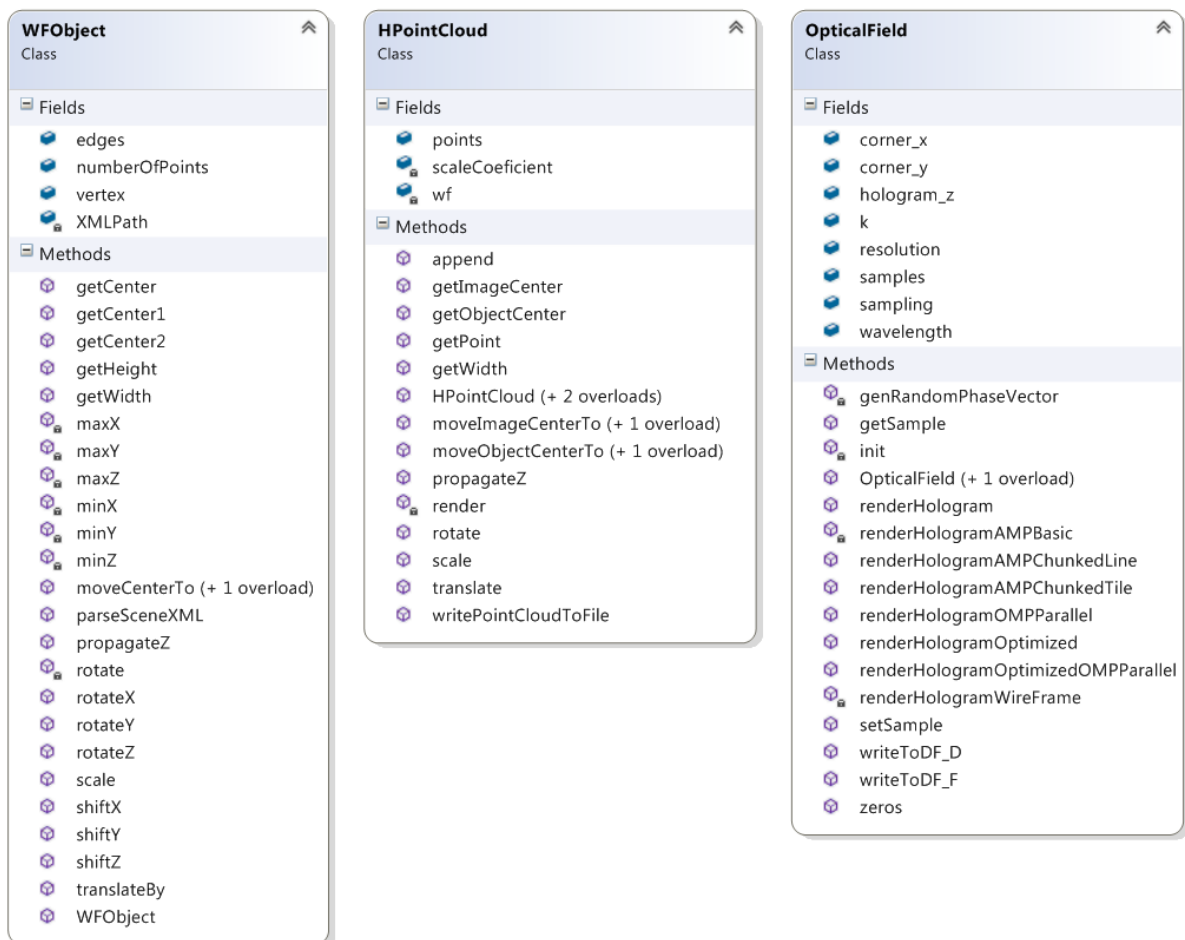
najvhodnejšia. Na základe popisu wireframe objektu sú potom rozpoznané steny a podľa špecifikovaných parametrov je obrázok umiestnený na jednu, prípadne viacero z nich. Následným spojením vznikne finálna množina bodov, z ktorej je po vhodných transformáciách – aby bol objekt vo vhodnej pozícii a veľkosti voči hologramu – možné jednoducho syntetizovať optické pole.

Zo zadaných konfiguračných parametrov je vytvorený objekt reprezentujúci optické pole, a to je zvolenou metódou vypočítané a uložené. Výstupmi programu sú dva súbory: spočítané optické pole vo formáte DF a voliteľný súbor v generickom formáte obsahujúci množinu bodov objektu.

Overenie správnosti výpočtu prebieha pomocou HoloToolkitu. Program DFToHologram zo vstupného DF súboru vytvorí hologram vo formáte PNG tým, že k objektovej vlne uloženej v optickom poli pripočíta referenčnú vlnu dopadajúcu pod špecifikovaným uhlom. Tento výsledok je zvyčajne ťažko interpretovateľný, preto je potrebné ešte využiť program HoloPropagLarge. Jeho výstupom je numerická simulácia rekonštrukcie hologramu – ak je použitý správne, na výstupnom PNG obrázku by mala byť vstupná scéna, samozrejme vrátane modifikácií vykonaných programom.

5.2 Významné časti riešenia

V tejto podkapitole sú popísané najdôležitejšie časti riešenia v logickom poradí ako nasledujú za sebou pri behu programu.



Obrázok 5.2: Niektoré významné triedy aplikácie.¹⁸

¹⁸ „Class diagram“ vygenerovaný Visual Studiom.

Spracovanie XML a wireframe model

Na spracovanie obidvoch vstupných XML súborov je využitá knižnica `rapidxml`¹⁹, ktorá umožňuje jednoduchú, odľahčenú a rýchlu prácu s týmto formátom. Wireframe objekt je v programe reprezentovaný vlastnou triedou (`WFObjekt`), ktorá obsahuje tabuľku hrán aj tabuľku vrcholov. Pri spracovaní vstupu príslušná funkcia zabezpečuje prechádzanie súboru po jednotlivých uzloch a naplnenie oboch tabuliek.

Prevod na point cloud

Wireframe model predstavuje informáciu o tom, ktorý bod je spojený s ktorým. Jednotlivé body reprezentuje trieda `Point3D`, ktorá má zložky `x`, `y`, `z` pre každú súradnicu trojrozmerného euklidovského priestoru. Pri prevode na point cloud reprezentáciu (trieda `HPointCloud`), čiže na množinu bodov, je žiaduce medzi jednotlivými vrcholmi vygenerovať príslušný počet bodov ležiacich na tejto úsečke. Zároveň musí platiť, že sú rozmiestnené rovnomerne. Princíp riešenia je nasledovný:

```
PointCloud pc;
float sumaHran := WireFrame.suma_dlzok_vsetkych_hran();
foreach Hrana h ∈ WireFrame.tabulkaHran do
    Point3D A := h.zaciatokHrany();
    Point3D B = h.koniecHrany();
    int n := Scena.pocetBodov * vzdialenost(A,B) / sumaHran;
    float krok := 1.0 / n;
    Point3D C = B - A;
    float t = 0.0;
    for i:= 0 to n do
        float x = A.x + C.x * t;
        float y = A.y + C.y * t;
        float z = A.z + C.z * t;
        t := t + krok;
        pc.vlozBod(Point3D(x, y, z));
    end for
end for
```

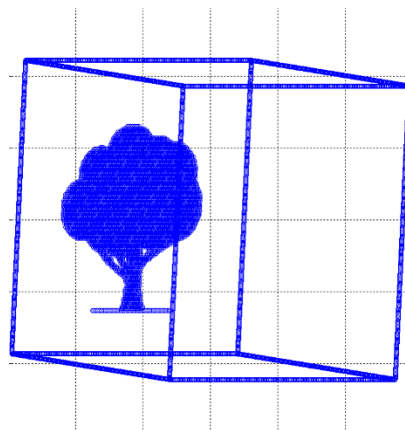
Algoritmus 5.1: Generovanie bodov na priamke s konštantným krokom.

Uvedený algoritmus pomocou parametrickej rovnice priamky analyticky počíta súradnice bodov s príslušným prírastkom. Pre každú hranu – úsečku – modelu sa vypočíta jej podiel z celkovej dĺžky všetkých hrán a k tomu príslušný podiel z celkového počtu bodov. Celkový počet bodov, z ktorých sa má skladať point cloud, je definovaný užívateľom vo vstupnom súbore popisujúcom model.

Trieda `HPointCloud` obsahuje aj funkciu na zápis množiny bodov do súboru. Vďaka tomu je možné za pomoci matematického softvéru, napríklad programu Matlab, vyniesť body na grafu a overiť ako tento model v skutočnosti vyzerá ešte predtým, než začne náročný výpočet hologramu.

¹⁹ Dostupná na: <http://rapidxml.sourceforge.net/manual.html>

Zároveň tento postup umožňuje vizuálne overiť vzhľad scény. Príklad výstupu (už aj po pridaní obrázku, vid'. nasledujúca časť) z Matlabu je na obrázku 5.2.



Obrázok 5.3: Grafické zobrazenie bodov point cloud-u.²⁰

Pridanie obrázku na povrch objektu

Vstupný obrázok vo formáte BMP si možno predstaviť ako dvojrozmerné pole pixelov. Aby bolo možné tento obrázok využiť pri generovaní hologramu, musí byť prevedený na množinu bodov. Tieto body nemajú definovanú farbu, ako je to pri pixeloch obrázka, ich atribútmi sú len súradnice. Preto je obrázok nutné najprv binarizovať. Použitá bola metóda prahovania („thresholding“), ktorá je síce primitívna, ale na tieto účely postačujúca. Prevod farebného obrázka na odtiene šedej bol vykonaný pomocou empirického vzorca pre vzťah farby a intenzity[17]:

$$I = 0,299 * R + 0,587 * G + 0,114 * B, \quad (5.1)$$

Potom, pomocou vhodne zvoleného prahu prebehla konverzia na binárny, čiernobiely obrázok. Odtiaľto je už veľmi blízko k množine bodov – stačí vybrať len biele (resp. čierne) pixely a k dvom súradniciam, x a y pridať konštantnú z súradnicu, napr. $z = 0$. Na prístup k pixelom obrázku vo formáte BMP, ich RGB zložkám a modifikáciu bola využitá knižnica EasyBMP²¹.

Poslednou prekážkou je rozdielna veľkosť a poloha v priestore. Riešením rozdielnej veľkosti je jednoduchá transformácia množiny bodov – zmenšenie alebo zväčšenie podľa veľkosti objektu. Riešenie polohy je zložitejšie. Množina bodov reprezentujúca obrázok leží v jednej rovine. Steny objektu tiež predstavujú body ležiace v jednej rovine avšak s rozdielnym umiestnením a rotáciou v priestore.

Základným krokom vôbec je rozpoznanie stien modelu. Keďže práca je zameraná najmä na primitívne útvary (štvorboký ihlan, kocka, kváder, atď.), zaujímavým obecným riešením je pokladať wireframe model za neorientovaný graf v zmysle diskkrétnej matematiky a vyhľadať v ňom elementárne cykly. Použitie riešenie je založené na metóde prehľadávania do hĺbky (DFS) a pri komplexných útvaroch by bolo časovo náročné. Pre potreby tejto práce je však dostatočné.

Ďalej je potrebné obrázok otočiť a posunúť. Po prevode na point cloud ležia všetky body automaticky v rovine xy . Potrebný uhol, o ktorý treba množinu bodov otočiť okolo osi x , určuje uhol,

²⁰ Výstup z programu Matlab.

²¹ Dostupná na: <http://easybmp.sourceforge.net/>

ktorý zvierá rovina xz s rovinou steny modelu. Uhol dvoch rovín v radiánoch sa vypočíta jednoducho podľa vzťahu:

$$\alpha = \cos^{-1}(n_1 * n_2 / |n_1| * |n_2|), \quad (5.2)$$

kde n_1 a n_2 predstavujú normálové vektory rovín. Obdobný postup je aplikovaný na rotáciu okolo osi y . Umiestnenie na stenu modelu je vykonané ďalšou transformáciou – obrázok je posunutý do bodu, ktorý je vypočítaným stredom steny. Veľkosť posunutia určuje rozdiel stredného bodu steny a stredného bodu obrázku.

Po vykonaní týchto akcií je scéna kompletne prevedená na množinu bodov, ktoré budú reprezentovať jednotlivé bodové zdroje pri syntéze optického poľa.

Optické pole

Implementované v triede `OpticalField`. Ide o akýsi obal optického poľa – okrem dvojrozmerného poľa komplexných čísel, ktoré predstavuje objektovú vlnu, obsahuje informácie potrebné na syntézu, napríklad vlnovú dĺžku svetelného zdroja či vzorkovanie. Zároveň sa tu nachádza niekoľko metód syntézy optického poľa. Všetky vychádzajú z algoritmu 2.1 popísaného v kapitole Holografia.

Základná implementácia (`renderHologram`) využíva priamo uvedený algoritmus bez žiadnych optimalizácií alebo urýchlenia. Na reálne účely je nepoužiteľná, jej úloha je čisto ilustračná a pre porovnanie urýchlenia.

Druhým variantom je paralelná verzia (`renderHologramOMPParallel`). Využíva direktívu `OpenMP`, pomocou ktorej je cyklus `for` rozdelený medzi viacero vlákien. Pri paralelnom behu vlákien je nutné pre správny výsledok zabezpečiť, aby si navzájom neprepisovali medzivýsledky. Riešenie je v tomto prípade triviálne – stačí vymeniť v spomínanom algoritme poradie cyklov – najprv bude priechod všetkými bodmi optického poľa a pre každý z nich sa potom bude počítat' kompletný výsledný príspevok od všetkých bodových zdrojov.

Implementovaný bol tiež optimalizovaný variant (`renderHologramOptimized`) aj s jeho paralelnou verziou (`renderHologramOptimizedOMPParallel`). Optimalizácia spočíva v princípe popísanom v kapitole Digitálna holografia, časť Techniky optimalizácie, keď sa snaží znížiť počet náročných matematických operácií pomocou aproximácie vzdialeností bodov. Dôležitým parametrom je práve veľkosť štvorca optického poľa, v rámci ktorého sa tieto vzdialenosti aproximujú. Vhodná veľkosť bola nájdená empiricky, podrobnosti viď. sekcia Testovanie.

```
rozdeleneBZ := rozdel_na_casti(BodoveZdroje, velkostCastiBZ);
castiHologramu = rozdel_na_casti(Hologram, velkostCastiHolo);
foreach castHolo ∈ castiHologramu do
    foreach castBZ ∈ rozdeleneBZ do
        pripocitaj_prispevok(castHolo, castBZ);
    end for
end for
```

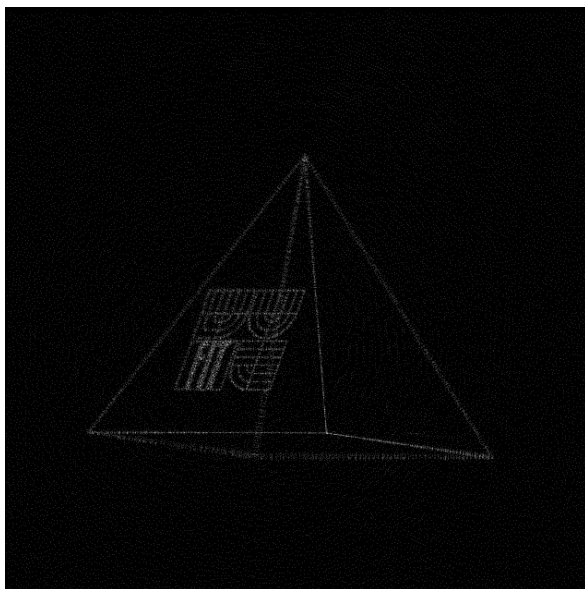
Algoritmus 5.2: Syntéza optického poľa na GPU s prevenciou TDR.

Posledné dve verzie využívajú technológiu C++ AMP a pomocou nej akcelerujú výpočet na GPU. Potrebné dáta sú skopírované do kontajnerov, s ktorými vie grafický procesor pracovať, nastaví sa rozsah výpočtov a s využitím paralelného cyklu prebehne výpočet. Objavuje sa tu však niekoľko

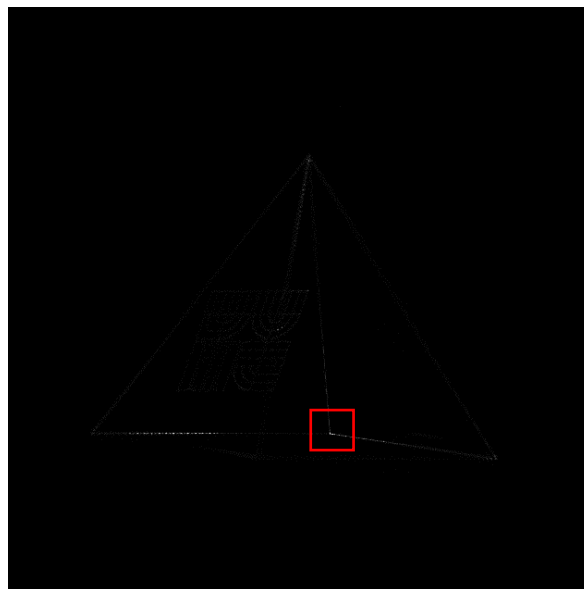
komplikácií. Problémom je napríklad nedostatočná veľkosť pamäte grafickej karty pri výpočtoch nad veľkým objemom dát. Problematickým sa javí aj mechanizmus TDR²². V systémoch Windows kontroluje odozvu grafickej karty a v prípade vypršania limitu (rádovo niekoľko sekúnd) ju zresetuje. Pri využívaní grafického procesoru na výpočty dlhšie ako tento limit sa výpočet preruší, čo je nežiaduce. Riešením oboch komplikácií je rozdelenie objemu dát, ktoré treba vypočítať, na časti. To platí ako pre bodové zdroje, tak aj pre časti optického poľa. Princíp ilustruje algoritmus 5.2.

Ďalšou záležitosťou, na ktorú treba brať ohľad je podpora dátového typu double na grafickej karte, väčšina starších kariet ho nepodporuje a výpočty sú obmedzené na presnosť typu float. Taktiež používanie zložených dátových typov nie je možné. Pre počítanie s komplexnými číslami bolo potrebné rozdeliť výpočty reálnej a imaginárnej zložky.

Verzia `renderHologramAMPChunkedTile` používa rozdelenie na štvorce. Druhá verzia, `renderHologramAMPChunkedLine` rozdeľuje optické pole po riadkoch. Toto rozdelenie je výhodné najmä z dôvodu toho, že je minimalizovaná pamäťová náročnosť a vypočítané výsledky sú priamo zapisované do súboru bez ukladania do poľa. Formát súboru DF totiž udáva, že jednotlivé komplexné čísla sú uložené štýlom riadok za riadkom. Výsledkom je stav, kedy je v jeden moment v pamäti len tá časť optického poľa, ktorá sa počíta.



Obrázok 5.4: Scéna s homogenizovanou intenzitou.²³



Obrázok 5.5: Scéna bez homogenizácie intenzity. Označený je presvetlený bod.²³

Všetky uvedené metódy využívajú funkciu `genRandomPhaseVector()`. Tá vráti pole naplnené pseudonáhodnými hodnotami z intervalu $\langle 0, \pi \rangle$ - pre každý bodový zdroj jedna. Hodnota vyjadruje počiatočnú fázu v danom bode. Cieľom je homogenizácia intenzity výsledného hologramu. Bez tohto opatrenia by sa mohla v určitom bode hologramu zložiť mohutná konštruktívna (deštruktívna) interferencia a presvietiť ostatné body, t. j. tento bod samotný by bol pri numerickej rekonštrukcii veľmi svetlý a intenzita ostatných voči nemu by bola zanedbateľná. Miernou nevýhodou je vyššia hladina šumu v prípade homogenizovanej intenzity.

²² Timeout Detection and Recovery. Detailný popis dostupný na: <http://msdn.microsoft.com/en-us/library/windows/hardware/ff570087%28v=vs.85%29.aspx>

²³ Výstup programu `HoloPropagLarge`.

5.3 Testovanie aplikácie a výsledky

Testovanie prebiehalo formou numerickej rekonštrukcie vypočítaného optického poľa. Využité boli programy HoloToolkit-u - DFToHologram a HoloPropagLarge. S predpokladom, že tieto nástroje fungujú správne, a že výstupy približne zodpovedajú skutočnej rekonštrukcii, je možné overiť výsledky. Záverečná rekonštrukcia v optickom laboratóriu podrobila skúške tieto predpoklady.

Na začiatok je vhodné uviesť podmienky testovania. Z pohľadu porovnania výkonu sa uskutočnilo na dvoch strojoch. Prvým bol počítač, na ktorom prebiehal vývoj. Išlo o notebook s parametrami uvedenými v tabuľke 5.1. Druhým bol klasický stolový počítač v CVT FIT²⁴, parametre sú v tabuľke 5.2.

Operačný systém	Windows 7 64-bit
CPU	Intel Core i5 460M 2,27 GHz
Počet jadier / počet vlákien CPU	2 / 4
Operačná pamäť RAM	4 GB
GPU	ATI Mobility Radeon HD 5650
Veľkosť pamäte GPU	1024 MB

Tabuľka 5.1: Hardvérové parametre testovacieho počítača č. 1

Operačný systém	Windows 7 64-bit
CPU	Intel Core 2 Duo E6750 2,66 GHz
Počet jadier / počet vlákien CPU	2 / 2
Operačná pamäť RAM	2 GB
GPU	Intel GMA 3100
Veľkosť pamäte GPU	256 MB

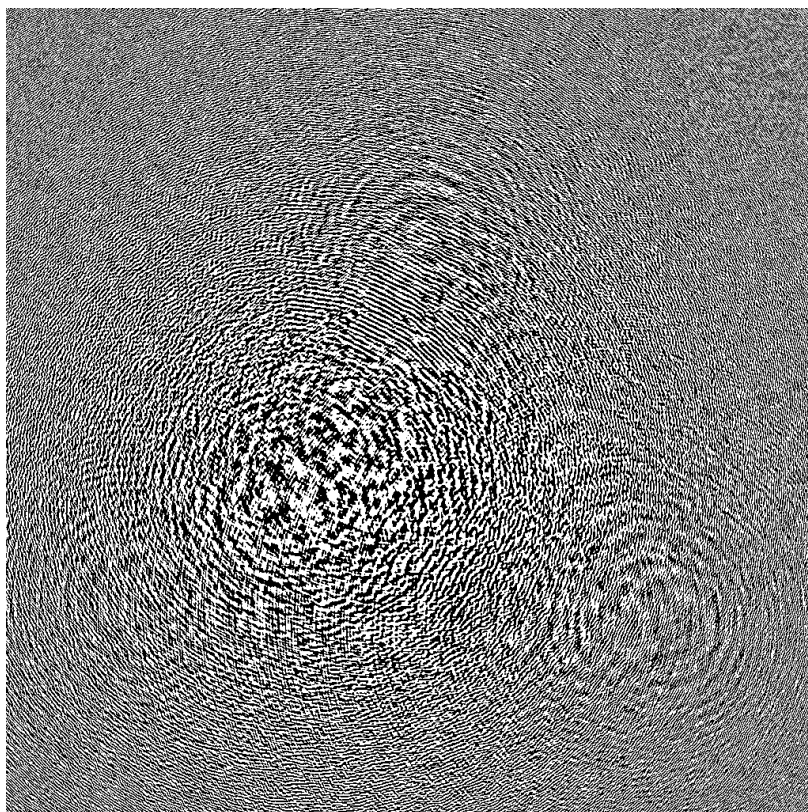
Tabuľka 5.2: Hardvérové parametre testovacieho počítača č. 2

Ešte je potrebné uviesť parametre pre testovanie generovania hologramov nastavené v konfiguračnom súbore. Rozlíšenie hologramu je 1024 x 1024, resp. 2048 x 2048 bodov. Veľkosť jedného bodu je 20 μm . Z modelu bolo vygenerovaných približne 4500 bodov point cloud-u. Scéna sa skladá zo štvorbokého ihlanu a loga FIT na jednej zo stien. Uvedené nastavenie je však vhodné len pre potreby testovania a porovnania časov, v skutočnosti by išlo o veľmi malý a nekvalitný hologram.

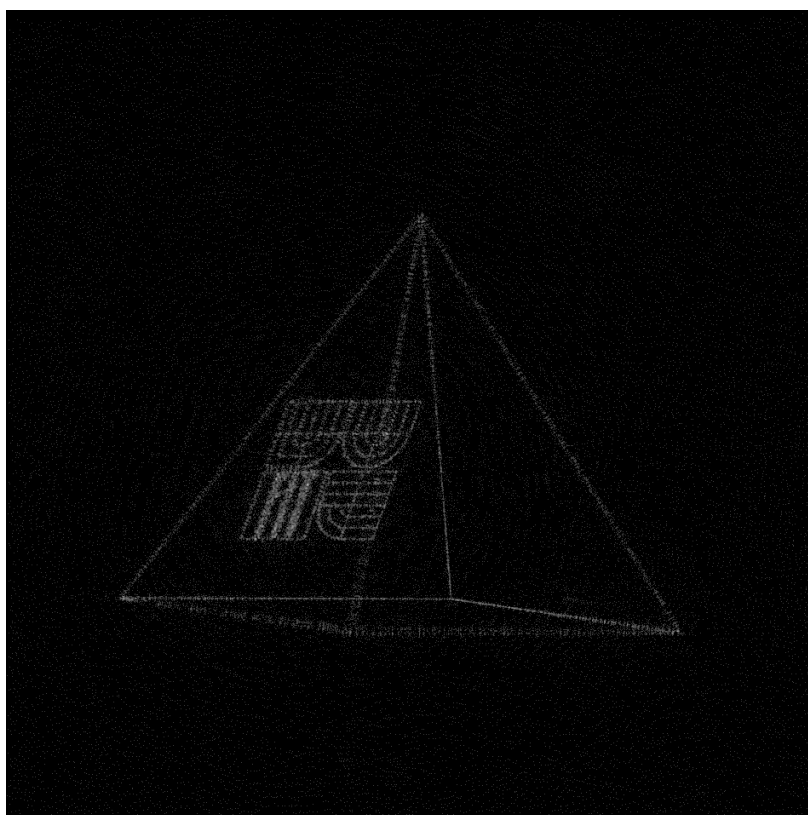
Primárnym výstupom programu je optické pole vo formáte DF. To predstavuje záznam objektovej vlny. Referenčná vlna bola pripočítaná programom DFToHologram, ktorého parametre je možné nastaviť v príslušnom XML súbore. Obrázok 5.6 ukazuje práve takýto výstup.

V nasledujúcom kroku bol pomocou programu HoloPropagLarge tento hologram numericky zrekonštruovaný. Parametre je opäť možné nastaviť v príslušnom XML súbore, dôležitými hodnotami sú predovšetkým propagačná vzdialenosť, ktorá by mala reflektovať vzdialenosť objektu od roviny hologramu, či rozlíšenie. To nesmie byť nižšie ako rozlíšenie hologramu. Výstup je na obrázku 5.7.

²⁴ Centrum výpočetní techniky Fakulty informačních technologií v Brně, počítačová učebna



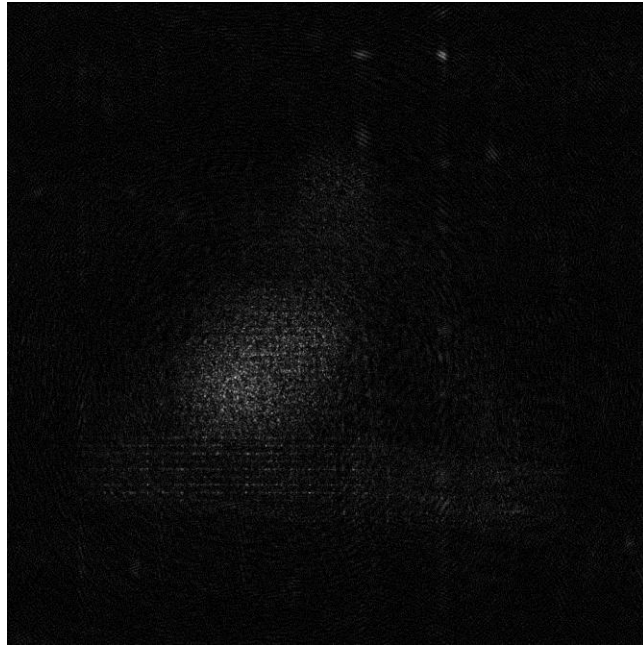
Obrázok 5.6: Binarizovaný hologram. Výstup programu DFTtoHologram.



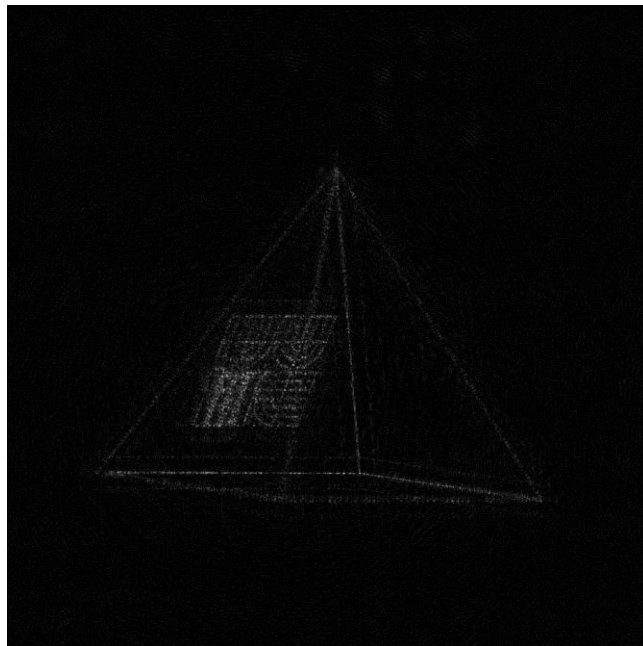
Obrázok 5.7: Numerická rekonštrukcia hologramu. Výstup programu HoloPropagLarge.

Keď bola takýmto spôsobom overená funkčnosť riešenia pre všetky metódy, bolo možné pokračovať smerom k dvom cieľom. Prvým cieľom bolo nájsť optimálnu veľkosť štvorca, v rámci ktorého dochádza k aproximácii vzdialeností pri optimalizovanej metóde syntézy hologramu. Podrobnejší popis optimalizácie je v kapitole 2.3. Na základe jej fungovania je logickým predpokladom to, že so zväčšujúcim sa rozmerom štvorca sa skracuje čas výpočtu, ale znižuje kvalita. Bolo potrebné nájsť vhodnú kombináciu.

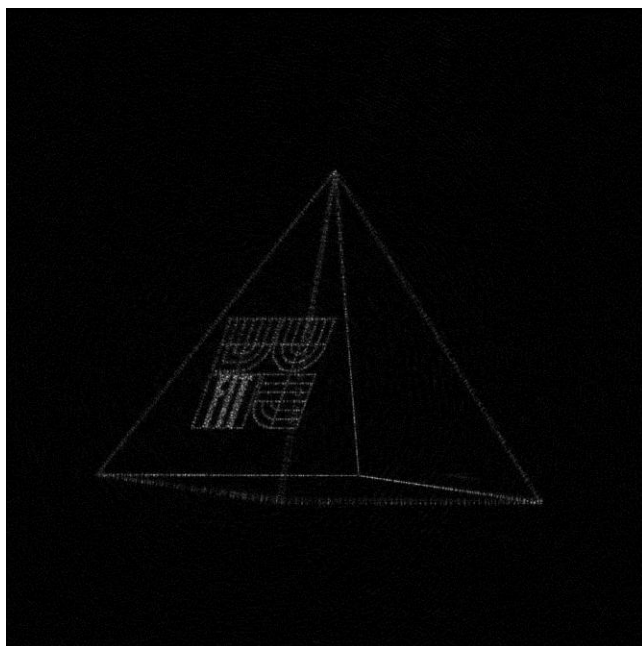
Testovanie prebiehalo na testovacom počítači č. 1, pri rozlíšení hologramu 1024×1024 . Nasledujúce výstupy numerickej rekonštrukcie vypočítaného hologramu ilustrujú postup. Veľkosť štvorca bude označená symbolom D , čas výpočtu optického poľa (v sekundách) T .



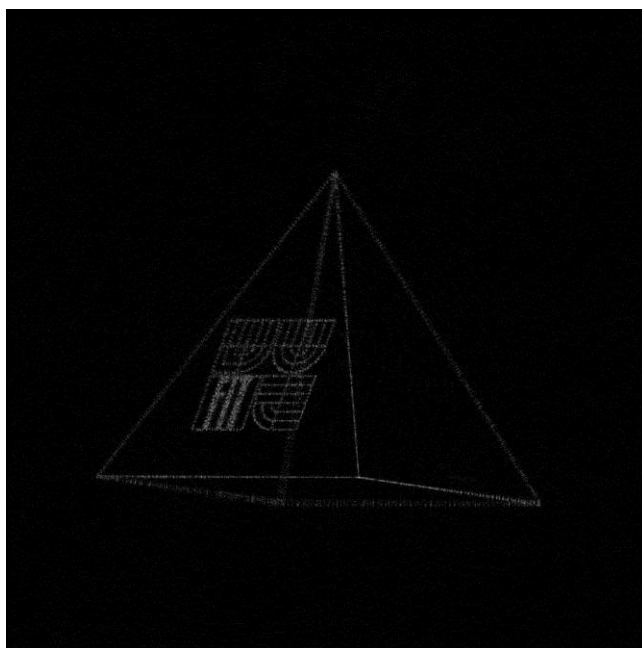
Obrázok 5.8: Numerická rekonštrukcia hologramu, $D = 128$, $T = 42$ s



Obrázok 5.9: Numerická rekonštrukcia hologramu, $D = 64$, $T = 70$ s



Obrázok 5.10: Numerická rekonštrukcia hologramu, $D = 32$, $T = 92$ s



Obrázok 5.11: Numerická rekonštrukcia hologramu, $D = 16$, $T = 106$ s

Na základe uvedených výstupov a časov je možné dospieť k záveru, že ideálna veľkosť štvorca je pri daných podmienkach $D = 32$, pretože oproti nastaveniu s parametrom $D = 16$ nie je badateľné zlepšenie kvality, ale na druhej strane je evidentný pomalší čas výpočtu, približne o 13 %. Ďalej bude v testovaní používaná verzia s nájdeným najvýhodnejším nastavením veľkosti štvorca.

Druhým cieľom bolo porovnanie výkonu jednotlivých implementácií a zhodnotenie prínosu optimalizácie. Každá metóda bola spustená päťkrát, merané boli čisto časy výpočtu optického poľa bez ostatných operácií. Extrémy boli odstránené a výsledný čas je aritmetickým priemerom zostávajúcich hodnôt. Nasledujúce dve tabuľky porovnávajú časy pre rôzne varianty syntézy.

Implementácia	Čas výpočtu	
	1024 x 1024	2048 x 2048
Klasická, jednovláknová	878 s	3447 s
Klasická, paralelná	579 s	2221 s
Optimalizovaná, jednovláknová	56 s	215 s
Optimalizovaná, paralelná	8 s	31 s
GPU	5 s	19 s

Tabuľka 5.2: Porovnanie časov behu syntézy na konfigurácií č. 1.

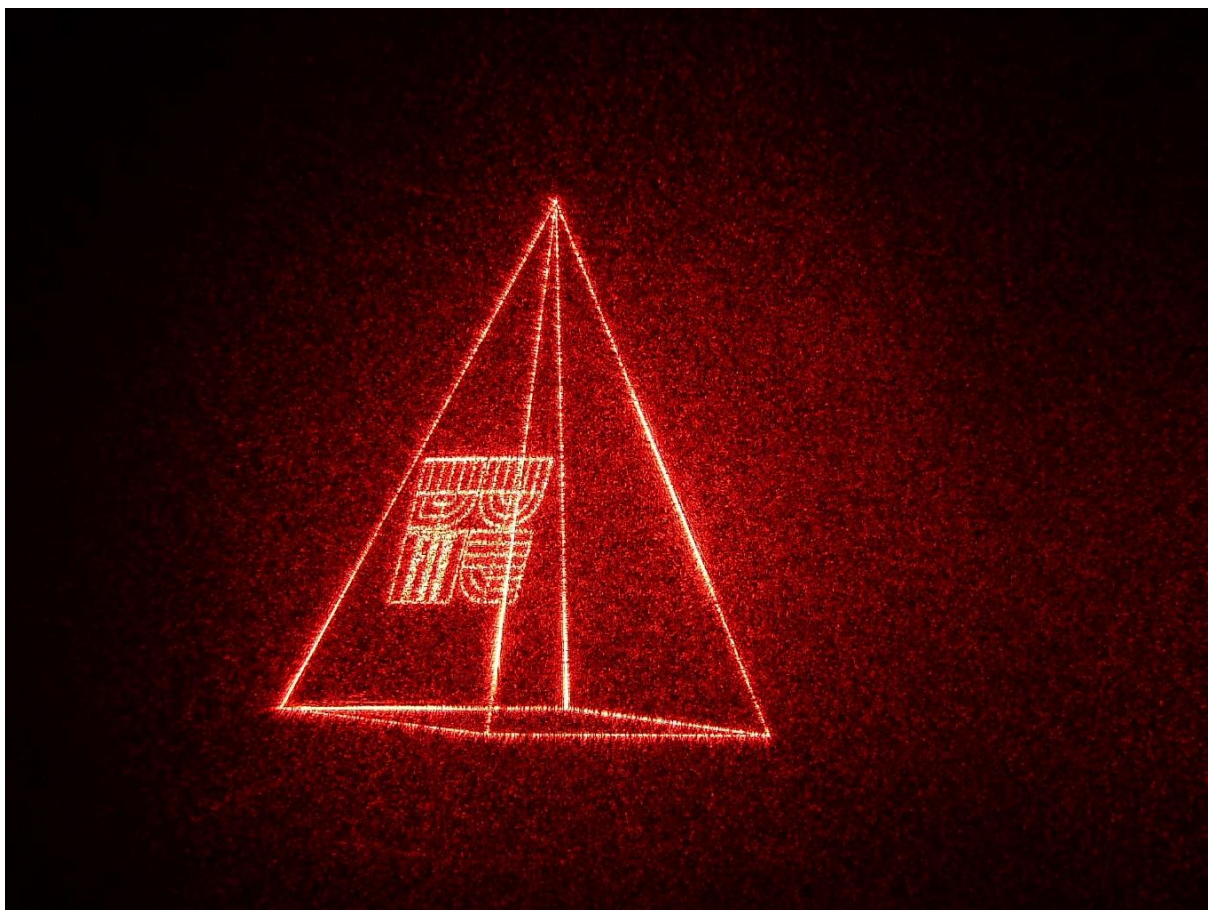
Implementácia	Čas výpočtu	
	1024 x 1024	2048 x 2048
Klasická, jednovláknová	1059 s	4239 s
Klasická, paralelná	624 s	2421 s
Optimalizovaná, jednovláknová	44 s	206 s
Optimalizovaná, paralelná	12 s	46 s
GPU	107 s	423 s

Tabuľka 5.2: Porovnanie časov behu syntézy na konfigurácií č. 2.

Z uvedených výsledkov je možné vyvodiť niekoľko záverov. V prvom rade, klasická metóda, t. j. bez žiadnej optimalizácie, je skutočne vhodná len na demonštratívne účely a ani paralelizácia na procesore nie je veľkým prínosom. Na druhej strane, zníženie počtu náročných operácií pri optimalizovanom variante rapidne zlepšilo časy, pri konfigurácií č. 1 približne 10-krát, pri konfigurácií č. 2 dokonca 14-krát oproti naivnej verzii. Paralelizácia optimalizovanej varianty je taktiež podstatným krokom vpred, napriek tomu, že beží len v rámci CPU dochádza k približne štvornásobnému urýchleniu pri druhej konfigurácii. Pri prvej ide až o hodnotu 7, čo je pravdepodobne spôsobené neznámou anomáliou pri testovaní. Keďže počas testov prišlo k poruche tohto stroja, pravú príčinu sa nepodarilo odhaliť.

Akcelerácia na GPU v prípade počítača č. 1 priniesla očakávaný výsledok, a to neporovnateľný nárast výkonu oproti ostatným verziám, s výnimkou paralelnej optimalizovanej, tu je nárast len zhruba 1,5-násobný. To sa však netýka počítača č. 2, hoci aj tento výsledok sa dal očakávať, keďže používaná grafická karta je integrovaná a nevýkonná, oproti karte z prvej konfigurácie je rozdiel niekoľko tried.

Posledným testom bola rekonštrukcia hologramu vyrobeného z vypočítaného optického poľa. Rekonštrukcia prebehla podľa plánov v optickom laboratóriu na FIT VUT v Brne. Napriek neľahkým podmienkam sa podarilo zostaviť optickú sústavu a úspešne vykonať potrebné náležitosti. Na obrázku 5.11 je fotografia virtuálneho obrazu, ktorý bolo možné pozorovať. Reálny obraz sa napriek snahe nepodarilo nájsť.



Obrázok 5.11: Fotografia virtuálneho obrazu z rekonštrukcie v optickom laboratóriu.²⁵

²⁵ Fotografia je dodatočne upravená kvôli názornosti, pri rekonštrukcii bola vertikálne prevrátená.

6 Záver

Cieľom práce bolo implementovať a porovnať jednotlivé varianty vybranej metódy syntézy hologramu, pomocou metódy vygenerovať hologram zvoleného wireframe modelu a zrekonštruovať ho. Cieľ sa podarilo úspešne splniť.

Začiatok spočíval v prvom rade zo štúdia dostupnej literatúry, keďže oblasť holografie je vo svojej podstate fyzikálnou vednou oblasťou. Ako základný algoritmus som si zvolil výpočet difrakčného integrálu, ktorý je síce obecné pomalší, ale svojou povahou ideálny na paralelizáciu. Po implementácii metódy v základnom variante som mohol som pristúpiť k jej akcelerácii. Prvou možnosťou bola optimalizácia pomocou aproximácie vzdialeností, neskôr pribudla jej paralelná verzia v rámci CPU. Ďalej pre porovnanie bolo vhodné implementovať aj GPU variant, na čo som využil relatívne novú²⁶ technológiu C++ AMP. Všetky metódy vykazovali pri numerickej rekonštrukcii uspokojivé výsledky. Samotné programovanie prebiehalo viac menej hladko, keďže v súčasnosti existuje k OpenMP aj C++ AMP dostatok materiálov a samotné technológie sú zamerané na užívateľa. Niekoľko iterácií však samozrejme prebehlo.

Výsledkom je použiteľná aplikácia, ktorú je možné spustiť na viacerých platformách, hoci inom operačnom systéme ako Windows nie je možné využiť akceleráciu pomocou GPU. Z pohľadu porovnania metód sa ukázalo, že pri optimalizovanej metóde zníženie počtu operácií veľmi urýchlilo výpočet a je dokonca porovnateľné s rýchlosťou klasickej metódy bežiacej na grafickom procesore. Pre počítače s menej výkonnou grafickou kartou a bežným procesorom je dokonca rýchlejšou alternatívou. Poslednou aktivitou bola rekonštrukcia v optickom laboratóriu, ktorá sa vzhľadom na podmienky nadmieru vydarila a pozorovali sme veľmi pekný virtuálny obraz.

Základ pre rekonštruovaný hologram tvorilo vypočítané optické pole. Pri rozlíšení 8192 x 8192 bodov a vzorkovaní 2 μm , čo vo výsledku znamená hologram veľkosti 1,6 x 1,6 cm, a pri scéne skladajúcej sa zo 40-tisíc bodových zdrojov, trval výpočet približne 50 minút.

Práca mi ukázala fascinujúcu oblasť holografie a zdôraznila, že hoci výkon počítačov rastie míľovými krokmi, v niektorých oblastiach naráža na limity. Počítačom generovaná holografia je práve takou, pri predstave, koľko by trval výpočet plnohodnotného hologramu veľkosti napríklad štandardnej počítačovej obrazovky je namieste určitá skepsa, o videu nehovoriac. Vývoj však pokračuje, a tak ako Denis Gabor síce objavil holografiu, ale na skutočný pokrok sa čakalo až do vynájdenia laseru, tak aj súčasné technické limity budú, nepochybne, prekonané.

V budúcnosti je možné pokračovať a nadviazať na túto prácu viacerými spôsobmi. Prvým aspektom je spríjemnenie práce používateľovi. Hoci primárne nie je aplikácia určená pre široké masy bežných užívateľov, bolo by prospešné pridať prehľadné grafické rozhranie. Túto ideu som síce rozpracoval, ale vytvorené rozhranie spomaľovalo výpočty a pre cieľ práce nebolo dôležité, takže do záverečnej verzie sa nedostalo. Rovnako by bolo zaujímavé upraviť možnosti vstupu, tak aby bolo možné pracovať aj so zložitými objektmi. Na druhej strane sa ponúka rozšírenie aplikácie o nové metódy syntézy hologramov. Z uvedených možností by som sa v budúcnosti rád zameral na grafické rozhranie aplikácie a pridanie nových metód syntézy.

²⁶ Technológia C++ AMP bola pre verejnosť uvoľnená v roku 2012.

Literatúra

- [1] BENTON, S. A.; BOVE, V. M. *Holographic Imaging*. Wiley-Interscience, 2008. ISBN 978-0470068069.
- [2] HARIHARAN, P. *Basics of holography*. Cambridge University Press, 2002. ISBN 978-0521807418.
- [3] JEONG, H. T. *What Are the Main Types of Holograms Today?* [online]. © 2005 Integraf LLC, [cit. 2014-02-23]. Dostupné z: <http://www.integraf.com/a-Types_of_Holograms.htm >.
- [4] LOBAZ, P. *Computer Generated Holography: 3D Vision and Beyond* [přednáška]. Západočeská univerzita v Plzni, Univerzitní 8, 30614 Plzeň, September 2013.
- [5] SENDERÁKOVÁ, D. Holografia – skutočne 3D obraz. *Jemná mechanika a optika*, 2007, roč. 52, č. 7-8, s. 229-231. ISSN 0447-6441.
- [6] HANÁK, I. *Urychlení výpočtu číslicového hologramu*. Plzeň, 2009. 142 s. Dizertační práce na Fakultě aplikovaných věd Západočeské univerzity v Plzni na katedře informatiky a výpočetní techniky. Vedoucí diplomové práce prof. Ing. Václav Skala, CSc.
- [7] JANDA, M. *Digital Hologram Synthesis*. Technical Report No. DCSE/TR-2007-02, Department of Computer Science and Engineering, University of West Bohemia, Univerzitní 8, 30614 Plzeň, Czech republic, December 2006.
- [8] REICHL, J.; VŠETIČKA, M. *Encyklopedie fyziky* [online]. © 2006 – 2014, [cit. 2014-02-21]. Dostupné z: <<http://fyzika.jreichl.com/main.article/view/431-optika>>.
- [9] KASPER, J. E.; FELLER S. A. *The Complete Book of Holograms: How They Work and How To Make Them*. Wiley, 1987. ISBN 978-0486415802.
- [10] *Holography Toolkit: Nástrojová sada pro digitální holografii* [online]. Fakulta aplikovaných věd Západočeské univerzity v Plzni, 2008- . [cit. 2014-04-23]. Dostupné z: <http://www.kiv.zcu.cz/en/research/downloads/product-detail-en.html?produkt_id=31>.
- [11] Microsoft Corporation. *C++ AMP: Language and Programming Model* [online]. 2012, [cit. 2014-05-12]. Dostupné z: <<http://download.microsoft.com/download/2/2/9/22972859-15C2-4D96-97AE-93344241D56C/CppAMPOpenSpecificationV12.pdf> >.
- [12] GREGORY, K.; MILLER, A. *C++ AMP: Accelerated Massive Parallelism with Microsoft Visual C++*. O'Reilly Media, 2012. ISBN 978-0735664739.

- [13] CHANDRA, R.; DAGUM, L.; KOHR, D.; MAYDAN, D.; McDONALD, J.; MENON, R. *Parallel Programming in OpenMP*. Morgan Kaufmann Publishers, 2001. ISBN 1558606718.
- [14] CHAPMAN, B.; JOST, G.; VAN DER PAS, R. *Using OpenMP: Portable Shared Memory Parallel Programming*. MIT Press, 2008. ISBN 978-0262533027.
- [15] HEDGE, S. *Solid Modeling Concepts*. Lecture of Computer Aided Design (AML710). Indian Institute of Technology in Delhi, Hauz Khas, 110 016 New Delhi, 2012.
- [16] EBERT, D. S.; MUSGRAVE, K. F.; PEACHEY, D.; PERLIN, K.; WORLEY S. *Texturing and Modeling: A Procedural Approach*. Academic Press, 1998. ISBN 978-1558608481.
- [17] ŠPANĚL, M.; HULÍK, R. *Geometrické transformace ve 2D a 3D*. Prednáška z predmetu Základy počítačové grafiky (IZG). Fakulta informačních technologií, Vysoké učení technické v Brně, Brno, 2013.
- [18] RUSU, R. B.; COUSINS, S. 3D is here: Point Cloud Library (PCL). *IEEE International Conference on Robotics and Automation (ICRA)*, 9-13 May 2011, Shanghai, China. ISSN 1050-4729.
- [19] HANÁK I.; ZEMČÍK P.; ŽÁDNÍK M.; HEROUT A. Hologram synthesis accelerated in field programmable gate array by partial quadratic interpolation. *Optical Engineering*, 2009, roč. 8, č. 48, s. 1-7. ISSN 0091-3286.